

Desenvolvimento de um manipulador para varreduras angulares com transdutores EMAT

João Pedro Torres de Sousa Andrade

**Desenvolvimento de um manipulador para
varreduras angulares com transdutores EMAT**

Aluno: João Pedro Torres de Sousa Andrade

Orientador: Alan Conci Kubrusly

Trabalho apresentado com requisito parcial à conclusão do curso de Engenharia de Controle e Automação na Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Brasil.

Resumo

Transdutores acusto-eletromagnéticos (do inglês, EMATs) geram ondas ultrassônicas, apresentando forte apelo prático para inspeções de meios metálicos. O desenvolvimento dos equipamentos e protocolos utilizados nessas inspeções necessita, geralmente, a realização de varreduras, ou seja, uma série de medições com variação espacial. Tais varreduras são procedimentos utilizados para acumular dados de maneira organizada, possibilitando o teste e comparação entre diferentes corpos de prova e configurações. Portanto, a facilidade e precisão com que se realiza tal varreduras é importante para o desenvolvimento dessas tecnologias de inspeção e ensaios não destrutivos.

Nesse trabalho foi desenvolvido um manipulador cujo objetivo é movimentar transdutores EMATs durante a realização de varreduras angulares. Essa automação requer desenvolvimento e integração de componentes mecânicos, eletrônicos e de software.

Este trabalho explora os conceitos básicos necessários a transdutores EMATs e as varreduras que os utilizam. Apresenta, a modelagem e construção do hardware mecânico, se aprofundando em alguns problemas encontrados, e na utilização de componentes impressos em 3D com tecnologia de modelagem por deposição de material fundido (do inglês, FDM). Desenvolve a eletrônica necessária para o funcionamento do projeto. Comenta a adaptação de software já existente às especificidades do manipulador e as varreduras que o utilizam. Por fim são apresentados alguns resultados obtidos com as ferramentas que foram desenvolvidas, comparando-as com o procedimento manual. Foi concluído que o manipulador funcionou de acordo com o esperado. Mesmo podendo ser ainda melhorado foi possível realizar varreduras de maneira consistente e com menos de um terço do tempo necessário manualmente. Além disso são apontadas recomendações de desenvolvimento futuro do projeto.

Palavras-chave: EMAT; varredura; manipulador; firmware; microcontrolador;

Development of a robotic angular manipulator for EMAT screening.

Abstract

Electromagnetic acoustic transducers (EMATs) generate ultrasonic guided waves which can be used in inspections. The development of equipment e protocols used in these inspections requires the realization of sweeps. These sweeps are procedures used to gather data in an organized way, making possible the test and comparison between different equipment and settings. Therefore, the easy with which these sweeps are made is key to the development of theses inspections technologies and non-destructive testing.

In this work a robotic manipulator was developed aiming to move these EMAT transducers while a sweep is being made. The automation of this procedure requires the integration of mechanical, electrical and software components.

This capstone project explores the basic concepts of EMAT transducers and the sweeps that uses them. The modeling and construction of mechanical hardware is discussed, with emphasis given to some problems that were encounter, and the utilization of components made with fused deposition modeling (FDM) 3D printing technology. It also presents the development of the electrical components need to the manipulator's operation and the adaptation of existing software to the project specificities. At last, some results gathered with the develop tools are shown and compared to manual acquisitions. The conclusion reached was that the arm worked as expected. Although there are some points to evolve, it was possible to reliably make screenings with less than a third of the usual necessary time. Other than that, some recommendations for future development are made.

Keywords: EMAT; Sweep; manipulator; firmware; microcontroller;

Lista de figuras

Figura 2-1- Esquemático dos componentes de uma varredura.	9
Figura 2-2 - (a) Direção de propagação das ondas na placa (b) Direção do movimento do transdutor na varredura.	10
Figura 3-1 - Esquemático Hardware Mecânico.	11
Figura 3-2 - Modelo do manipulador (a) Peças impressas em amarelo. (b) Divido em: estrutura central (roxo), braço (azul claro) e estrutura periférica (verde).	12
Figura 3-3 - Peça central do braço marcada em vermelho (a) na montagem cortada (b) isolada, parte superior (c) isolada, parte inferior.	13
Figura 3-4 - Modelo do eixo da saída da caixa de redução.	14
Figura 3-5 - - Fotografiografia da primeira iteração da peça central do braço. Área da falha no plástico circulada em vermelho.	14
Figura 3-6- Fotografiografia da segunda iteração da peça central do braço, mostrando parafuso torto.	15
Figura 3-7 - Fotografia da terceira e ultima iteração da peça central do braço.	15
Figura 3-8 - Modelo inicial (descartado) da caixa do EMAT móvel com molas de extensão.	16
Figura 3-9 - Modelo da caixa do EMAT móvel. Parafuso de fixação vertical circulado em vermelho.	16
Figura 3-10 - Modelo da caixa do EMAT móvel com molas de torção impressas.	17
Figura 3-11 - Organização de 'features' do Prusa Slicer (a) Legenda, (b) Caminho gerado, incluindo camadas superiores (c) Caminho gerado com corte vertical, possibilitando ver camadas de preenchimento.	17
Figura 3-12 - Caminho gerado pelo <i>slicer</i> (a) Utilizando apenas perímetros (b) Configurações padrão.	18
Figura 3-13 - Fotografia de ambas as peças impressas.	19
Figura 3-14 - Primeiro design (descartado) de movimentação vertical com servo motor.	19
Figura 3-15 - Fotografia do sistema de movimentação vertical.	20
Figura 3-16 - <i>Dragchains</i> indicadas por setas pretas. Linhas vermelhas indicando o corpo da <i>dragchains</i> , Setas vermelhas indicando movimentação do ponto móvel. Posição aproximada de 0° e 350mm de raio.	21
Figura 3-17 Algumas posições da <i>dragchain</i> central (a)~90°, (b)~135°, (c)~200°, (d)~360°.	22
Figura 3-18 - Medição de folga da junta central	23
Figura 4-1 - Esquemático geral da eletrônica.	24
Figura 4-2 - Esquemático do bloco de fontes.	24
Figura 4-3 - Fotografia da paca do microcontrolador com marcação nas conexões necessárias.	25
Figura 4-4 - Montagem da caixa da eletrônica (a) Modelo, sem tampa. (b) Fotografia da montagem final.	25
Figura 5-1 - Curva de velocidade de dois eixos.	27
Figura 6-1 - Fotografia EMAT construído no laboratório 22mm.	28
Figura 6-2 - Formato de um Ascan em diferentes posições angulares. Linha azul correspondente a 0°, vermelha a 30°, amarela a 44° e roxa a 90°.	29

Figura 6-3 - Diagrama de radiação (a) Varredura manual, passo 10° (b) Varredura automatizada, passo 2°	30
Figura 7-1 - Fotografia do sistema de fixação da caixa do EMAT receptor. Parafusos circulados em vermelho.	31
Figura 7-2 - Modelo da peça de montagem do EMAT periférico ressaltada vem vermelho.	31
Figura 7-3 - Fotografia do teste isolado do mecanismo de movimentação axial.	32

Sumário

1.	Introdução	8
2.	Processo de varredura	9
3.	Hardware mecânico	11
a.	Peça central do braço	13
b.	Posicionamento vertical do EMAT receptor.....	15
	O desenvolvimento dessas molas impressas:	17
c.	Gerenciamento dos cabos	20
d.	Jogo	22
4.	Eletrônica.....	24
5.	Software	26
a.	Firmware - Arduino	26
b.	Matlab® Script	27
6.	Resultados	28
7.	Conclusões e recomendações futuras.....	31
a.	Alterar maneira de fixar a caixa do EMAT receptor	31
b.	Movimentação vertical e servomotor	32
c.	Motorização do eixo radial.....	32
d.	Melhorar o jogo do eixo central.....	33
8.	Referências	34
	Apêndice A.....	36
	Apêndice B.....	43
	Apêndice C.....	45

1. Introdução

Ultrassom é útil em inspeção e sistema de monitoramento por ensaios não-destrutivos. É possível analisar a integridade física das estruturas observando os sinais de ondas ultrassônicas nelas propagantes. Defeitos e descontinuidades forçam que as ondas sejam refletidas, assim indicando sua presença. Além disso, esse processo, considerado um ensaio não destrutivo, pode ser realizado em campo, sem inviabilizar ou interromper o uso presente e futuro do equipamento testado [1] [2].

Para gerar e receber ondas de ultrassom, utiliza-se transdutores. Transdutores convencionais possuem elementos ativos do tipo piezoelétrico [1] [3]. Esses requerem contato com o meio em que se gera a onda, e é frequentemente utilizada uma camada de acoplante, líquido ou gel, para tal [2] [4]. Transdutores acústico-eletromagnéticos (EMAT, do inglês, '*electromagnetic acoustic transducers*'), por outro lado, são capazes de gerar e receber ondas ultrassônicas em meios condutores sem a necessidade de estarem em contato com o meio [5]. Além disso, EMATs são capazes de gerar diferentes tipos de ondas guiadas, e em especial ondas cisalhantes horizontais, cuja geração por transdutores convencionais é não-trivial [6] [7].

Varreduras consistem em várias medidas com pequenas variações de posição [8] [9] [10] [11]. Podemos comparar diferentes varreduras, no mesmo corpo de prova, para caracterizar e calibrar equipamentos, incluindo transdutores. Por isso, a precisão e facilidade com a qual essas varreduras são feitas é essencial para o desenvolvimento destas tecnologias de inspeção. É comum que estas varreduras feitas durante o processo experimental sejam realizadas à mão [8] [12]. Esse processo é laborioso e intrinsecamente menos preciso e mais demorado que métodos automatizados.

Os EMATs específicos que utilizaremos geram ondas guiadas longitudinalmente que se propagam perpendicularmente à face do transdutor. Por este motivo é necessário fazer varreduras angulares, uma necessidade incomum com utilizações mais tradicionais de ultrassom [12] [13] [14]. Utilizamos um transdutor no centro de uma chapa de metal e outro em vários pontos de determinado círculo em volta desse centro.

O projeto consiste em construir um manipulador com objetivo de automatizar estas varreduras. Para isso, utilizaremos coordenadas polares. Inicialmente o ângulo será variado de forma automatizada, e futuramente também o raio. O posicionador precisa ser comandado facilmente por um computador, e deve ser construído de maneira econômica, utilizando, em sua maioria, materiais já disponíveis.

O restante desse documento é organizado da seguinte maneira: No capítulo 0, é apresentado o processo de varredura angular, com os equipamentos e diferentes passos utilizados. No capítulo 0, é apresentada a construção do hardware mecânico, com ênfase no desenvolvimento de alguns pontos críticos. O capítulo 0 trata do desenvolvimento da eletrônica necessária. No capítulo 0 são identificadas as diferentes partes do software. O capítulo 6 apresenta alguns resultados obtidos. Por fim o capítulo 0 apresenta conclusões e enumera as recomendações de desenvolvimento futuro.

2. Processo de varredura

O processo de varredura se dá na repetição de medidas em diferentes posições. Cada medida possui vários passos que envolvem diferentes equipamentos. Na Figura 2-1 podemos observar os seis principais componentes e suas conexões necessárias.

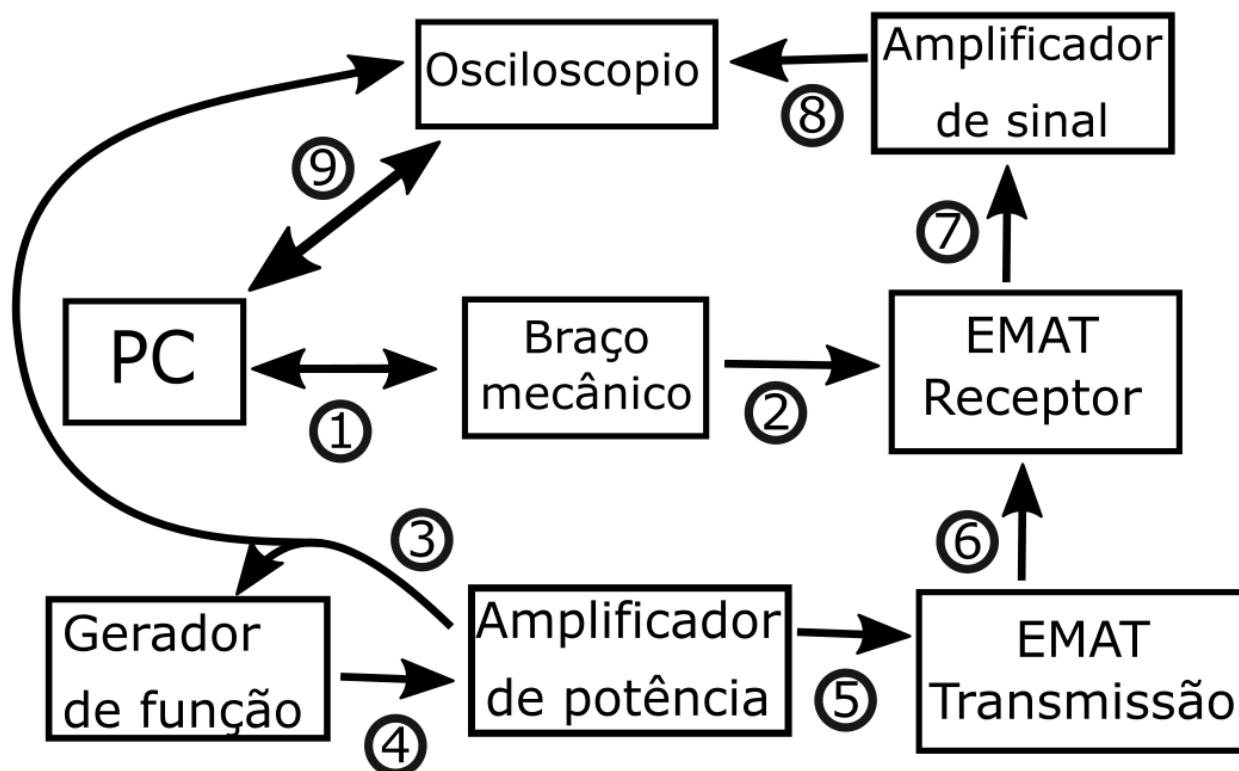


Figura 2-1- Esquemático dos componentes de uma varredura.

As diferentes conexões são:

1. Conexão entre o computador e o braço mecânico, via wifi, se comunicam com comandos específicos enviados pelo pedido de URLs.
2. Manipulação física do EMAT pelo braço.
3. Emissão de um trigger pelo Amplificador de potência que é recebido pelo Gerador de função e pelo Osciloscópio. Esse trigger sincroniza o processo e é emitido constantemente durante a varredura. Via cabo, e um splitter, BNC.
4. Emissão de um sinal, pulso, pelo gerador de função, via cabo BNC.
5. Geração do pulso que excita o EMAT de transmissão, via cabo BNC.
6. Propagação do sinal ultrassônico gerado pelo EMAT de transmissão até o receptor. Ocorre dentro do corpo de prova.
7. Geração de um sinal elétrico, de leitura do ultrassom, que o EMAT receptor envia ao amplificador de sinal via cabo BNC, que por sua vez repassa o sinal para Amplificador de sinal.
8. Sinal amplificado é repassado, via cabo BNC, para o Osciloscópio.
9. Pedido de aquisição por parte do PC e subsequente transmissão do formato de onda pelo Osciloscópio. Ocorre via Ethernet.

Em ordem cronológica, temos:

- Pedido de movimentação do braço (1).
- Execução do movimento, manipulação física do EMAT receptor (2).
- Confirmação de finalização do movimento pelo braço (1).
- Emissão do trigger (3).
- Emissão do pulso (4).

- Emissão do pulso amplificado (5).
- Propagação do ultrassom (6).
- Leitura do sinal gerado pelo EMAT (7).
- Transmissão do sinal de leitura amplificado (8).
- Recebimento dos dados da aquisição naquele ponto específico pelo computador (9).

Esse processo de varredura, gera um formato de onda, para cada ponto no qual foi feita uma medição. Esse formato de onda é chamado de Ascan e pode ser analisado para identificar características do corpo de prova, ou do próprio transdutor [15]. Podemos comparar varreduras, no mesmo corpo de prova, para calibrar os diversos equipamentos mencionados acima, principalmente o transdutor. Para isso, a precisão e repetibilidade da varredura é crítica. Além disso, ao diminuir o tempo e o esforço necessários para o procedimento, possibilitamos a captura de mais dados. A automação das varreduras, portanto, provê grande valor para o desenvolvimento dessas tecnologias.

O tipo específico de transdutores EMATs que utilizamos geram ondas guiadas. Essas ondas podem ser definidas como ondas de tensão, forçadas a seguir um caminho, predominantemente, definido ao longo de sua extensão [4] [16]. Em específico, são guiadas em uma placa de metal, se propagando nas direções indicadas na Figura 2-2 (a) de maneira não uniforme. Existem direções principais, nas quais a intensidade de propagação é a maior, e direções não-principais [12]. Um dos objetivos principais das varreduras é construir um diagrama de radiação que indica essa intensidade em cada direção. Para isso, como indicado na Figura 2-2 (b) utilizaremos o braço mecânico para movimentar o EMAT receptor em torno do transmissor, de modo que ele possa medir a transmissão das ondas em todas as direções.

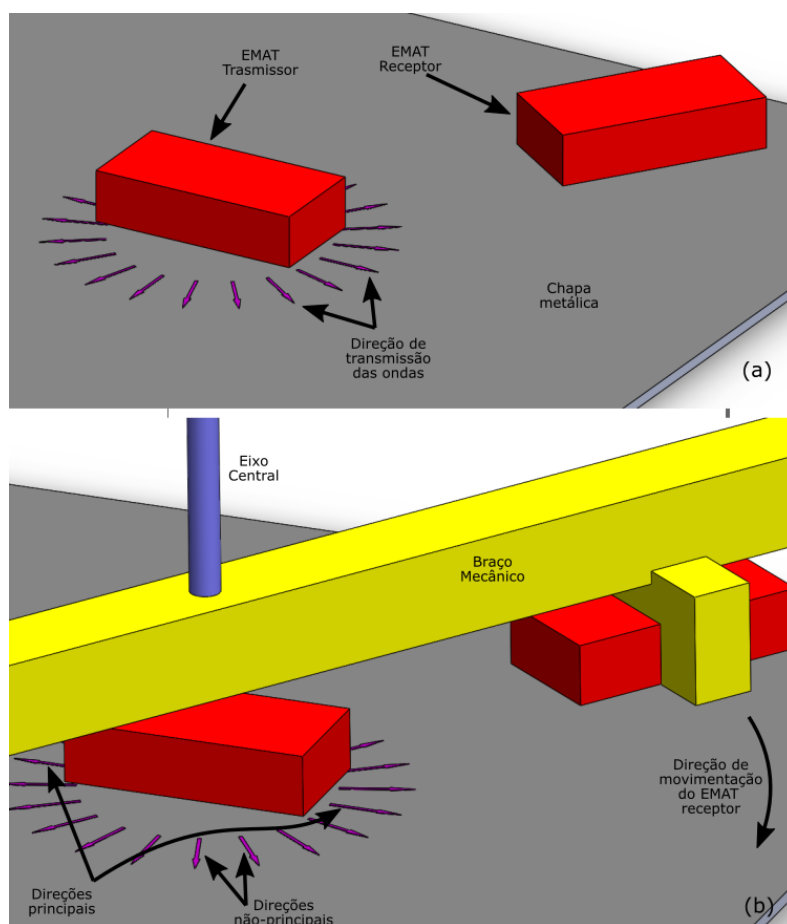


Figura 2-2 - (a) Direção de propagação das ondas na placa (b) Direção do movimento do transdutor na varredura.

Portanto, para automatizar esse tipo de varredura, precisamos de um hardware mecânico capaz de movimentar o EMAT receptor de maneira angular sobre uma chapa de metal.

3. Hardware mecânico

Dentre os requisitos básicos do hardware mecânico estão:

- A capacidade de segurar um transdutor EMAT Receptor e movimenta-lo de maneira angular sobre uma chapa de metal de forma automática.
- A possibilidade de movimentar o mesmo transdutor radialmente, mesmo que esse movimento não seja motorizado.
- O espaço para acomodação EMAT Emissor deve ser coincidente com o eixo de rotação.
- Organização de todos os cabos necessários.

O processo de desenvolvimento do hardware teve como objetivo criar uma estrutura relativamente simples, barata, fácil de modificar que atendesse às necessidades do projeto. Foram utilizados peças e materiais disponíveis no laboratório, assim como algumas peças doadas por mim. Por exemplo, ao invés de comprar um motor novo, de acordo com as especificações do projeto, foi utilizado um motor NEMA 23 disponível na sucata de um projeto antigo [17], e adaptado de acordo com as necessidades atuais, se utilizando uma redução, e regulando uma corrente menor do que a nominal do motor.

A estrutura foi feita em sua maior parte de perfis extrudados de alumínio e peças impressas em FDM (Fused Deposition Modeling). A impressora 3D, ferramenta muito utilizada no projeto, possibilita rápidas interações de diferentes designs, característica central ao processo de desenvolvimento do hardware. A modelagem das peças, assim como da montagem, foi feita em SolidWorks®.

Para atender os requisitos listados acima foi concebido uma estrutura indicada pelo esquemático da Figura 3-1 .

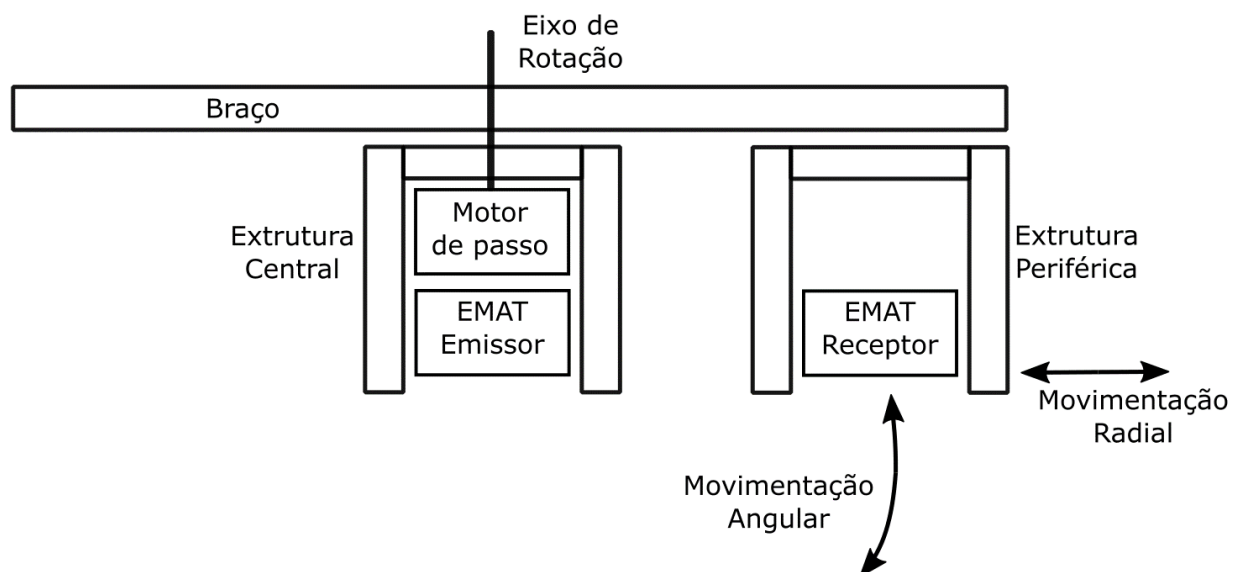


Figura 3-1 - Esquemático Hardware Mecânico

Depois de algumas iterações no desenvolvimento chegou-se ao modelo indicado abaixo na Figura 3-2 cujas partes são melhor explicadas no resto deste capítulo.

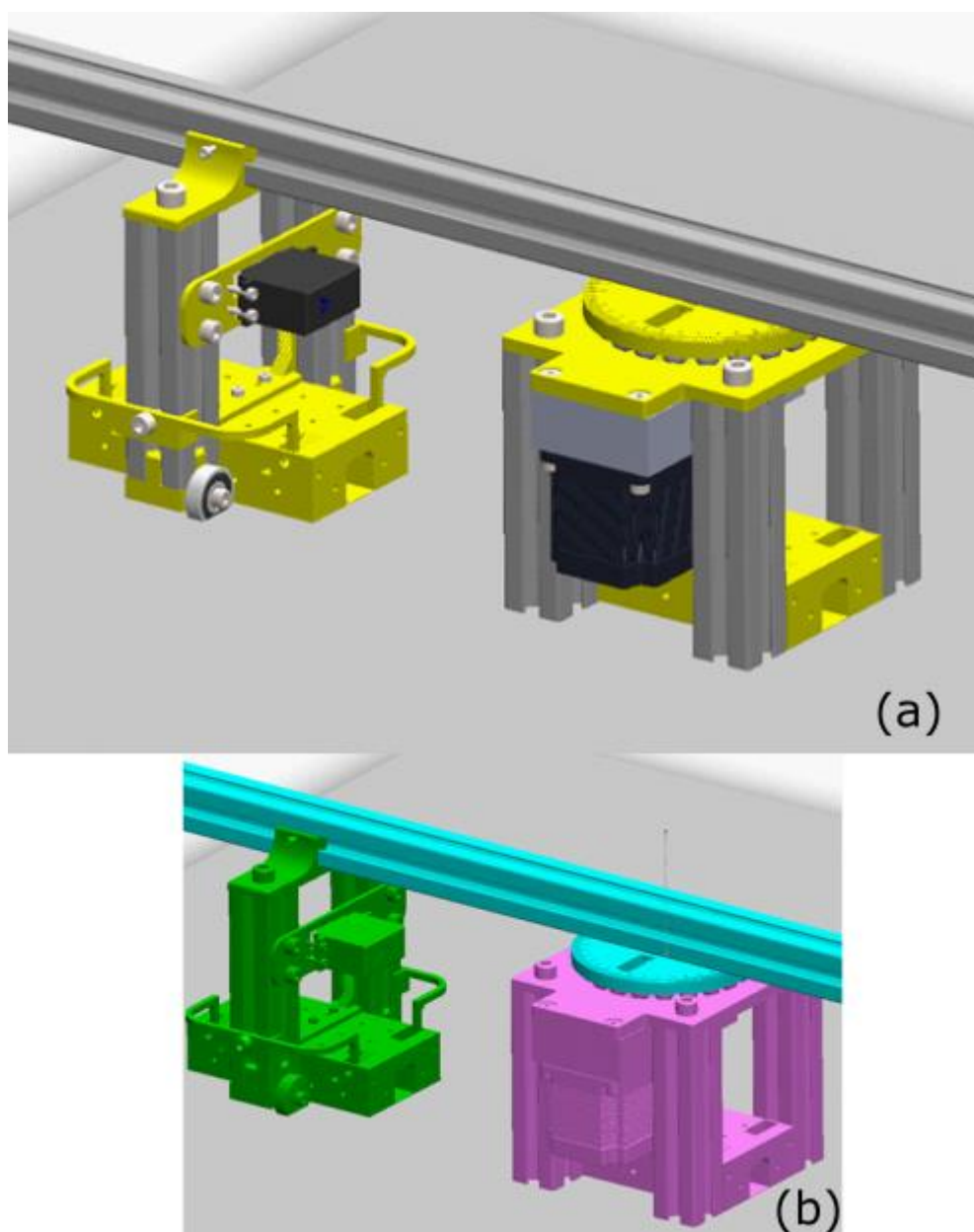


Figura 3-2 - Modelo do manipulador (a) Peças impressas em amarelo. (b) Divido em: estrutura central (roxo), braço (azul claro) e estrutura periférica (verde).

Podemos dividir a montagem em três partes de acordo com seu movimento. Essa divisão pode ser facilmente visualizada na figura Figura 3-2 (b).

A estrutura central (roxa) é estacionária. Ela é composta por quatro pernas feitas com perfis de alumínio, que se encaixam na caixa para um transdutor EMAT que fica no seu centro. A placa superior prende as pernas ao motor de passo e sua redução, e faz um rolamento com a placa do braço através de 20 esferas de aço.

O braço (azul) gira sobre o supracitado rolamento. O mesmo é composto por um longo perfil de alumínio, preso na estrutura central por sua peça central impressa.

A estrutura periférica (verde) gira juntamente com o braço e além disso desliza sobre o mesmo. Esse movimento que altera o raio da varredura, é feito manualmente e pode ser fixado apertando-se dois parafusos. Essa estrutura possibilita o movimento vertical da caixa do EMAT periférico, que desliza sobre os dois perfis de alumínio orientados verticalmente. Essa estrutura também possui molas e um motor de passo responsável por esse movimento.

As próximas subseções são dedicadas à explicação do processo de desenvolvimento do hardware, com ênfase aos pontos que apresentaram maior dificuldade, requerendo maior número de iterações e as soluções encontradas.

a. Peça central do braço

A peça central do braço possui vários requisitos. A mesma forma a parte superior do rolamento axial que suporta o braço através de uma calha na qual se encaixam as esferas. Além disso, é presa no perfil de alumínio do braço por meio de dois parafusos verticais. A mesma também possui marcações angulares, tal como um transferidor escolar, facilitando a leitura do ângulo do braço. Essa peça é ilustrada em vermelho na Figura 3-3.

Nesta subseção será discutido o fato de que essa peça precisa ser fixada ao eixo da redução (Figura 3-4) de maneira a transmitir o movimento giratório do mesmo para o braço. Esse encaixe foi escolhido como ponto de falha, no caso em que algo impeça a rotação principal do braço. Desse modo, temos uma peça relativamente barata, de fácil replicação e substituição, que protegerá componentes de maior custo. Essa é a mesma ideia por de trás da utilização de uma chave de segurança que age como uma espécie de fusível mecânico [18].

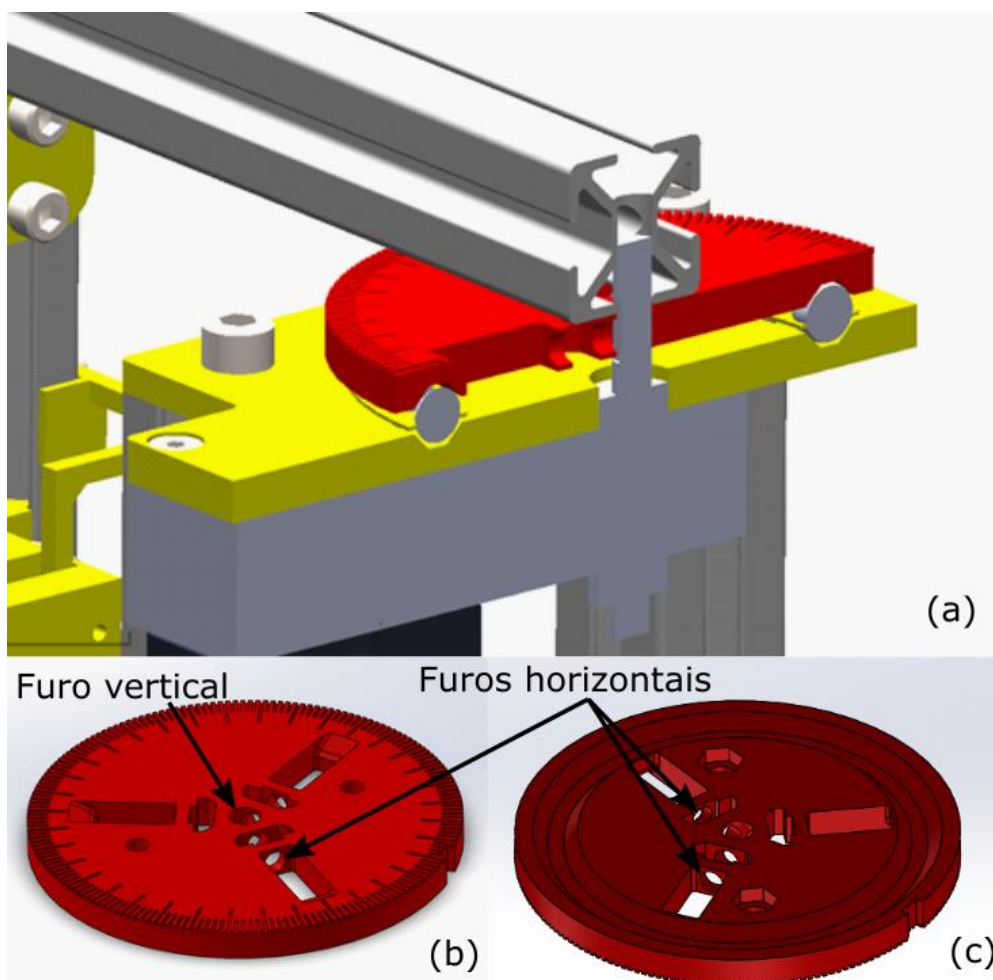


Figura 3-3 - Peça central do braço marcada em vermelho (a) na montagem cortada (b) isolada, parte superior (c) isolada, parte inferior.

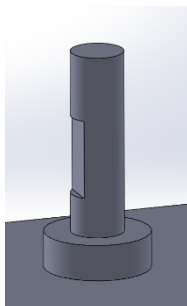


Figura 3-4 - Modelo do eixo da saída da caixa de redução.

Para esse formato do eixo, a chave não é o método normalmente utilizado, mas sim um parafuso perpendicular ao mesmo [18]. Vale comentar também que devido à parte superior do eixo ser redonda, o furo vertical não pode incluir a face plana do eixo. Inicialmente, foram utilizados três parafusos M3 com porcas, um para cada furo horizontal, um dos quais fica na face plana do eixo. Os outros dois parafusos, além de ajudarem a distribuir as forças pela peça, podem ser utilizados no evento de falha de um parafuso, trocando qual fica na face plana. Esses parafusos e porcas também são indicados na Figura 3-5. O custo do plástico da peça em questão, ignorando custos de energia e tempo de máquina, pode variar de R\$4 a R\$7 dependendo do tipo de plástico e configurações utilizadas. As diferentes iterações no design têm uma diferença mínima nesse quesito de custo.

A primeira peça feita funcionou corretamente nos primeiros testes. Contudo, a adição de mais componentes da máquina, como cabos e *dragchains*, motorização vertical e o próprio peso dos transdutores o esforço exercido pela junta principal aumentou. Uma falha ocorreu como esperado. A mesma pode ser observada na Figura 3-5 por uma descoloração característica do plástico Acrilonitrila Butadieno Estireno (do inglês ABS) no lugar onde o mesmo cedeu.

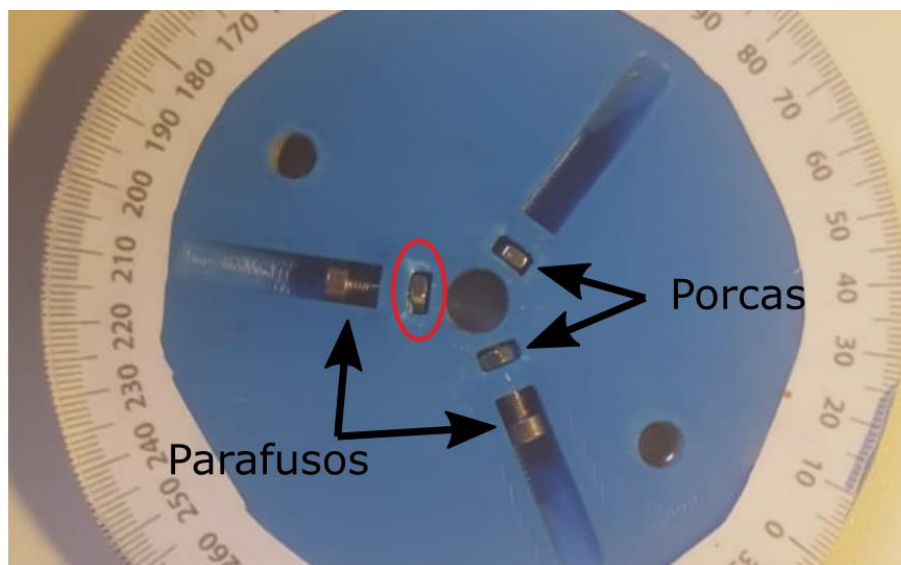


Figura 3-5 - - Fotografiografia da primeira iteração da peça central do braço. Área da falha no plástico circulada em vermelho.

Dado que a falha ocorreu durante o funcionamento normal da máquina foi preciso tomar algumas medidas para aumentar a força necessária para causar tal falha. A velocidade máxima de rotação foi reduzida e uma nova versão da peça foi feita. Com menor tolerância, maior preenchimento, especialmente na parte central, e utilizando polietileno tereftalato glicol (do inglês PETG) ao invés de ABS. Como resultado, o parafuso M3 falhou, enquanto a peça se manteve íntegra. Figura 3-6 .

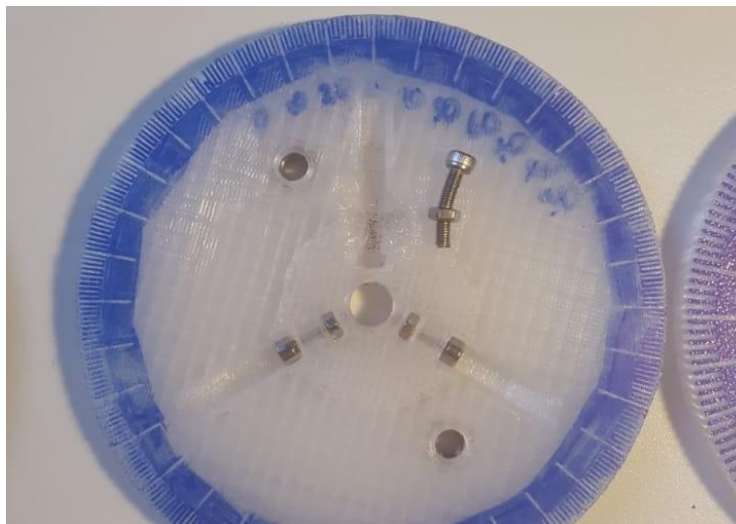


Figura 3-6- Fotografiografia da segunda iteração da peça central do braço, mostrando parafuso torto.

Na terceira iteração foi trocado o parafuso M3 por um M6. Utilizando uma porca T, conseguiu-se manter a peça com a espessura original de 8mm. Depois desta mudança, a peça funcionou como esperado falhando apenas quando o manipulador foi forçado de maneira incorreta. Esse processo de aumentar a resiliência da peça pode ser considerado, na analogia elétrica, equivalente a aumentar a especificações do fusível de forma a garantir que ele só queime quando o circuito estiver funcionando incorretamente. A Figura 3-7 exibe uma fotografia da peça com os parafusos e porcas utilizados em seu modelo final.



Figura 3-7 - Fotografia da terceira e ultima iteração da peça central do braço

b. Posicionamento vertical do EMAT receptor

Nessa subseção, são apresentados alguns passos do desenvolvimento do sistema de posicionamento vertical do EMAT receptor dentro da estrutura periférica.

Embora o EMAT não necessite estar em contato com o corpo de prova para leitura, é ideal manter uma distância mínima e consistente para os melhores resultados nas medições, essa distância é chamada de lift-off [19]. Por isso, ambas as caixas que contêm os transdutores possuem a parte inferior, doravante referida simplesmente como película, com 0.4 mm de espessura, dimensão escolhida em função da altura da camada de 0.2mm utilizada como configuração nas impressões. Foi almejado manter a parte inferior das caixas completamente em contato com a chapa.

Contudo, esse posicionamento vertical gerou um problema devido às modificações feitas no primeiro design concebido. Nele, a estrutura externa era mais similar à central, com quatro extrusões, conforme

ilustrado na Figura 3-8. Durante sua simplificação, foram eliminadas as quatro molas que de extensão que puxariam a caixa em direção ao chão.

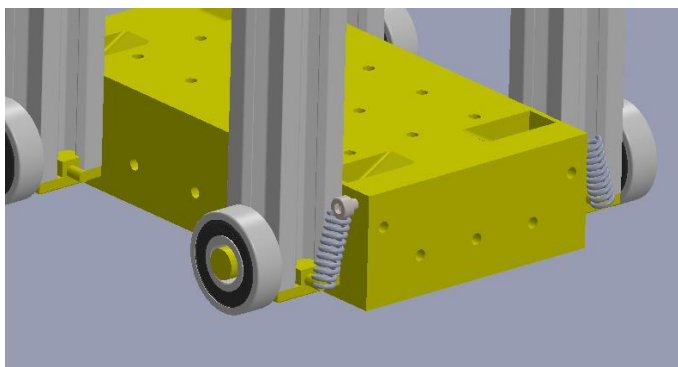


Figura 3-8 - Modelo inicial (descartado) da caixa do EMAT móvel com molas de extensão.

No design atual, a caixa da extremidade móvel pode ser deslocada verticalmente nos perfis de alumínio, e fixa com dois parafusos. Um dos lados é representado na Figura 3-9.

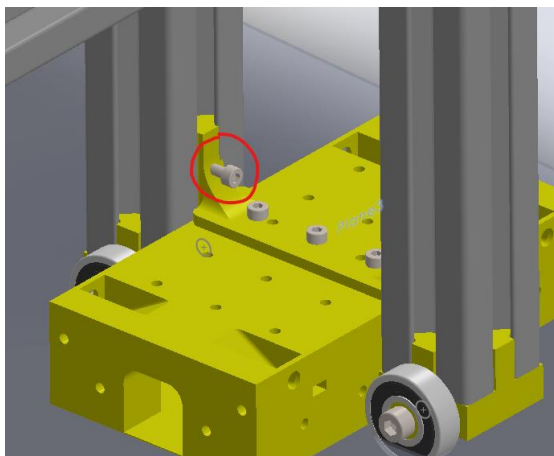


Figura 3-9 - Modelo da caixa do EMAT móvel. Parafuso de fixação vertical circulado em vermelho.

A estrutura de todo o manipulador foi construída de maneira rígida, com a mesma altura em ambos os lados central e periférico, de maneira com que o braço fique na horizontal. Devido a isso, teoricamente a estrutura central e periférica suportam seus respectivos pesos, e ambas dividem o peso do braço. Entretanto, pequenas diferenças em tolerância podem mudar drasticamente a distribuição de peso.

Foram utilizadas rodas na estrutura periférica para diminuir o atrito com a chapa. Pés de borracha e contrapeso no centro com objetivo oposto, de forma a manter o centro fixo enquanto o EMAT externo se move. Entretanto, caso a caixa do EMAT seja presa verticalmente, encostando-se no chão, de maneira que as rodas não encostem, o manipulador não funciona corretamente e a estrutura central começa a girar enquanto o braço fica parado.

A primeira tentativa de resolver esse problema consistiu em adicionar uma mola de forma que o peso suportado pela estrutura periférica pudesse ser dividido entre as rodas e a caixa do EMAT. Foram impressas molas com design inspirado na suspensão de carros de fórmula 1 [20]. Neles, a suspensão estilo 'push-rod' utiliza a torção de uma barra ao invés de uma mola tradicional. O design, demonstrado na Figura 3-10, permite que controlemos a pressão da caixa no chão. Caso o peso do próprio EMAT seja grande, também seria fácil adaptar a mola para "puxar" a caixa do chão. A regulação da mola pode ser feita movendo verticalmente o ponto de fixação da mesma sobre os perfis de alumínio.

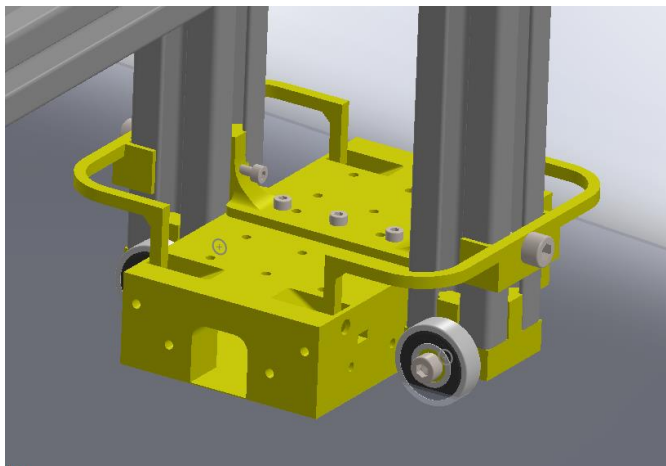


Figura 3-10 - Modelo da caixa do EMAT móvel com molas de torção impressas.

O desenvolvimento dessas molas impressas:

O desenvolvimento das molas impressas pode ser utilizado como bom exemplo para as idiossincrasias de peças impressas em FDM. Existem diferentes maneiras como podemos nos aproveitar de aspectos que, na maioria dos casos, são considerados desvantagens dessa tecnologia.

O primeiro modelo desse tipo de mola foi feito, como o restante da estrutura, com plástico ABS e configurações padrão de impressão. Embora a mola tenha funcionado, ela deforma muito pouco antes de exercer grande força. Ou seja, ela possui uma constante de elástica alta, e possui uma região de deformação plástica pequena. Isso torna a regulação da pressão sobre a caixa difícil.

Peças impressas em FDM consistem, basicamente de deposições de plástico fundido em linhas que constituem camadas, que por sua vez formam as peças. Um programa, comumente chamado de *slicer*, define o caminho pelo qual a impressora faz a peça. Esse caminho, com todas as linhas e camadas, define muitas características da peça [21]. É muito comum que peças impressas não sejam maciças, por isso o caminho é organizado em perímetros, superfícies inferiores ou superiores e preenchimento. Podemos ver a maneira como o *slicer* utilizado, Prusa Slicer, representa esses diferentes recursos na Figura 3-11.

Nesse exemplo, foram utilizadas dois perímetros e preenchimento padrão 'honeycomb' de 25%.

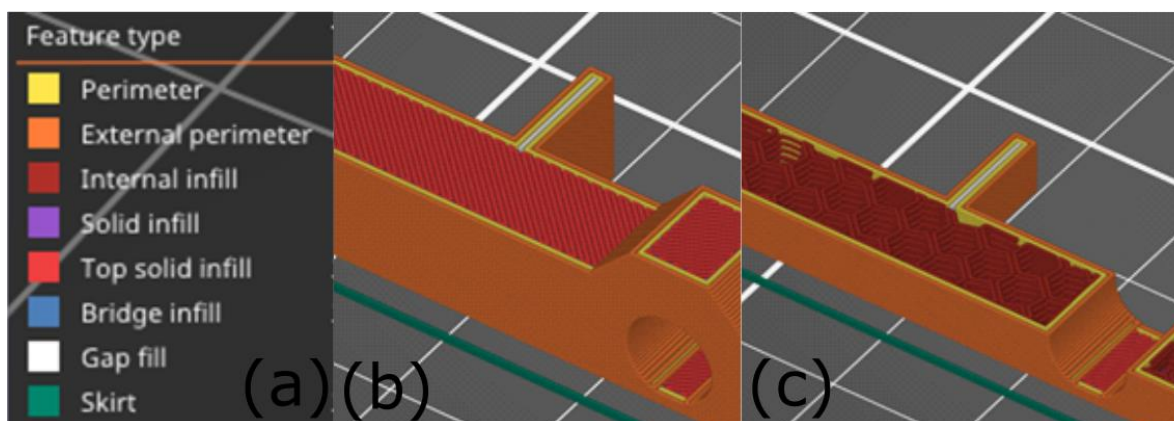


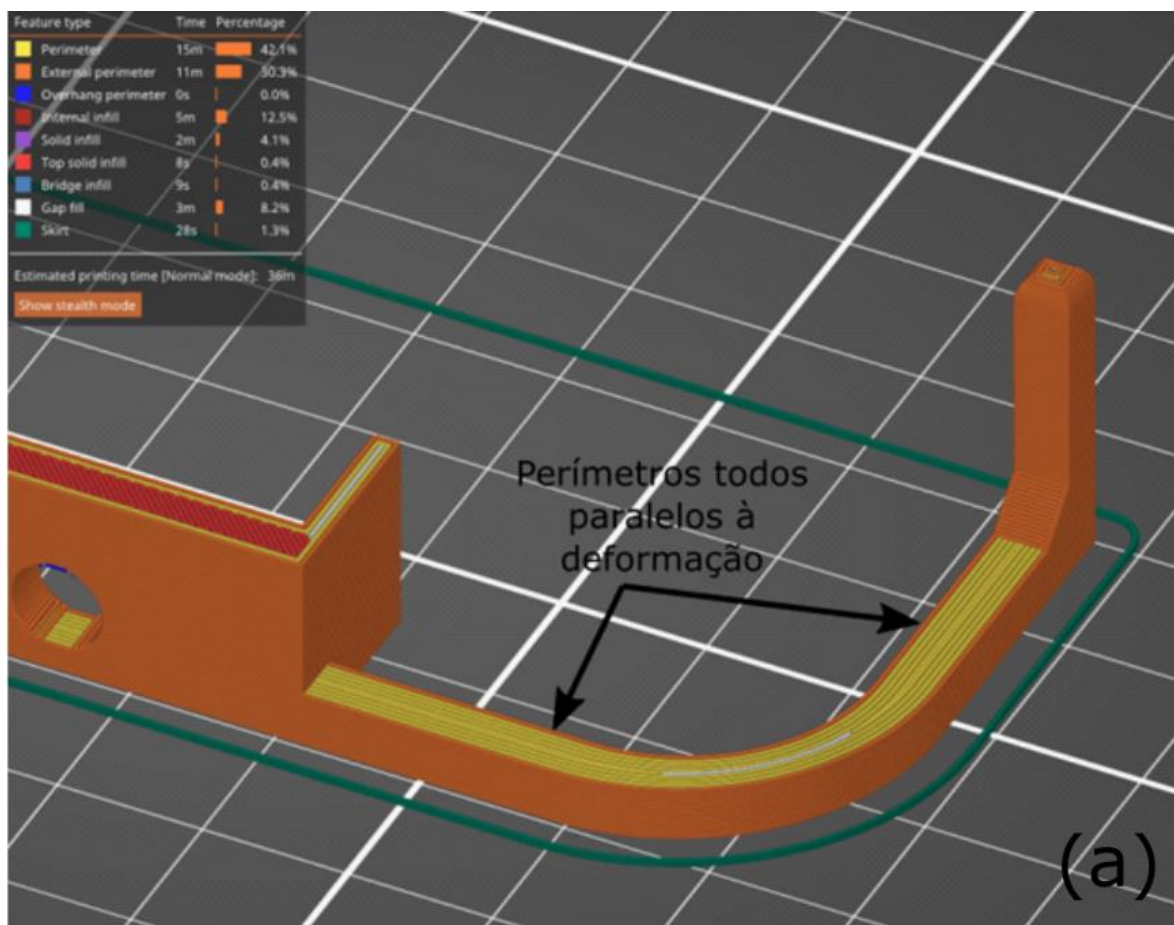
Figura 3-11 - Organização de 'features' do Prusa Slicer (a) Legenda, (b) Caminho gerado, incluindo camadas superiores (c) Caminho gerado com corte vertical, possibilitando ver camadas de preenchimento.

Peças impressas com tecnologia FDM são intrinsecamente anisotrópicas [22]. Além do plástico, impressora e modelo, as configurações de impressão possuem um enorme efeito no resultado. Para tornar a regulação do EMAT mais fácil tentou-se, empiricamente, promover maior deformação na mola, de maneira mais linear.

Para isso, o modelo foi alterado, principalmente diminuindo sua espessura. Foi utilizado plástico polilático (do inglês PLA), que é mais rígido, apesar de mais quebradiço. Nas configurações de impressão, apenas aumentando o número de perímetros conseguiu-se eliminar qualquer camada superior, inferior,

ou preenchimento. Dessa forma a peça se torna ainda mais anisotrópica, ela possui uma resistência muito grande ao longo das linhas de perímetro, sendo difícil de quebrar nesta direção. Contudo, devido à falta de camadas superiores, inferiores e preenchimento, ela também tem maior facilidade para dobrar na direção desejada. Podemos observar as configurações de impressão, e as peças finais na

Figura 3-12.



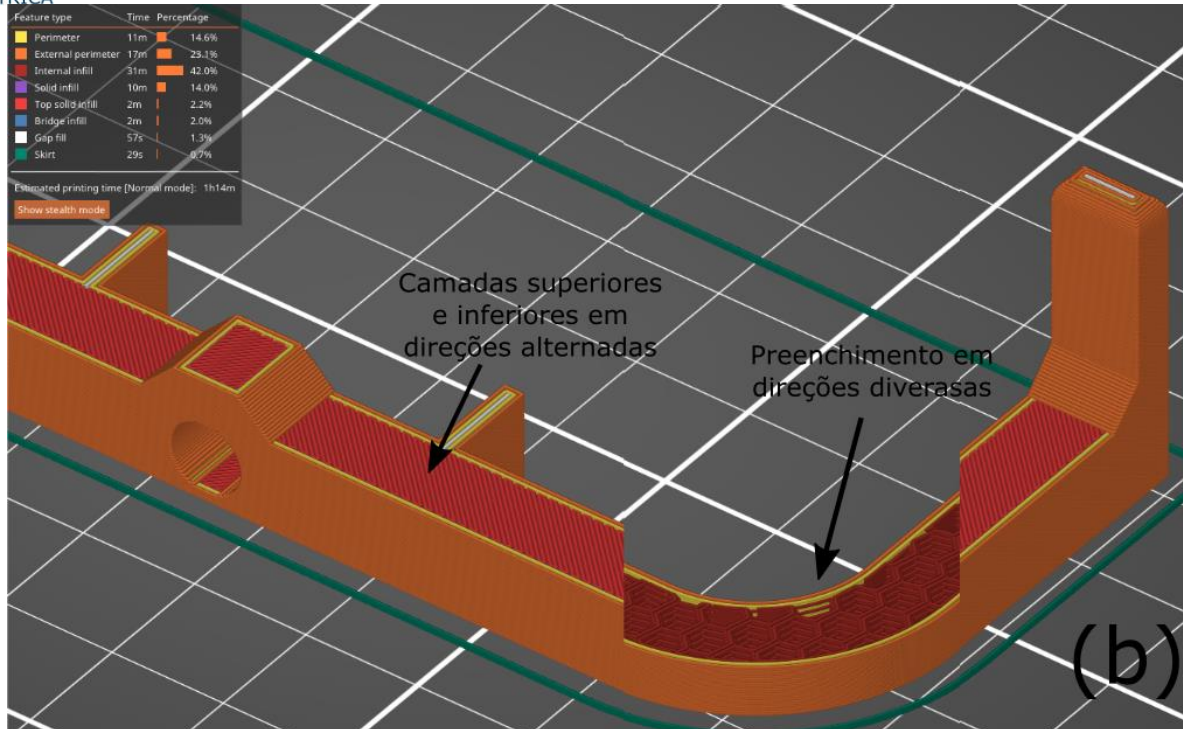


Figura 3-12 - Caminho gerado pelo *slicer* (a) Utilizando apenas perímetros (b) Configurações padrão

Os resultados da impressão da nova mola resolveram o problema de regulação. Na Figura 3-13, podemos ver o uma fotografia das duas peças impressas.

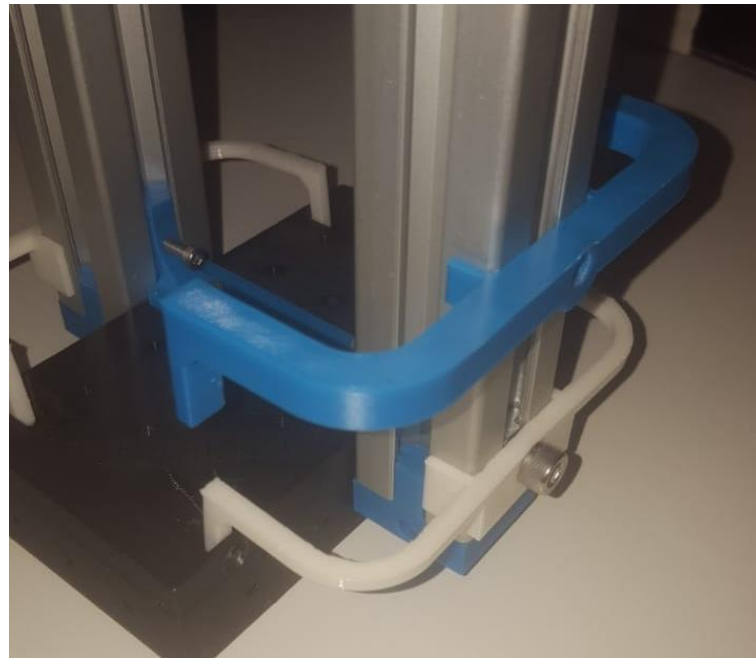


Figura 3-13 - Fotografia de ambas as peças impressas.

Contudo, mesmo depois do desenvolvimento das molas, a movimentação do braço ainda não ficou consistente. As inconsistências de algumas superfícies do corpo de prova fazem com que o braço, ao girar, saia da horizontal e mude sua distribuição de peso. O problema observado anteriormente, no qual a estrutura central girava, enquanto o braço e estrutura periférica ficavam paradas, se manteve de maneira intermitente.

Como solução definitiva, utilizamos um servo motor MG996R [23] para mover verticalmente a caixa do EMAT receptor entre cada passo. Dessa maneira, garantimos a consistência, tanto do movimento, quanto da distância entre o EMAT e o corpo de prova, durante a medição. Vale comentar que a medição só ocorre entre passos da varredura, com o manipulador estacionário na posição inferior.

O primeiro design concebido, mas ultimamente descartado, utilizava uma engrenagem com cremalheira. Ele pode ser observado na Figura 3-14.

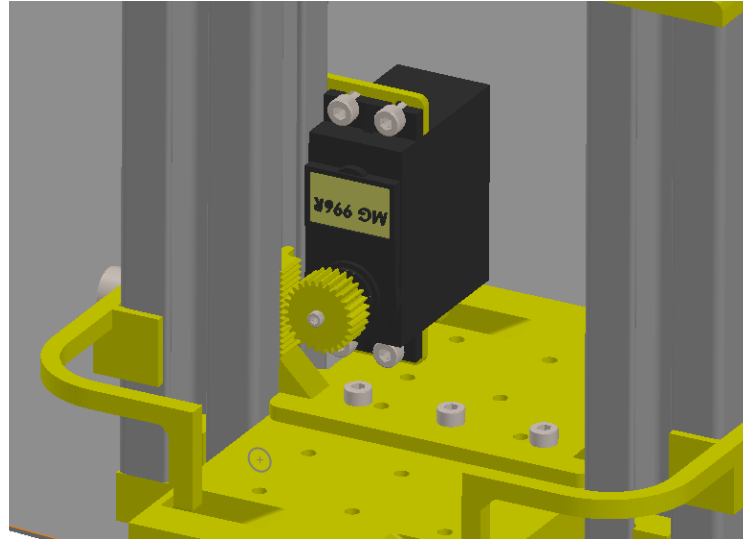


Figura 3-14 - Primeiro design (descartado) de movimentação vertical com servo motor.

O design final pode ser visto na Figura 3-15. O software foi modificado de maneira que o EMAT receptor seja içado, contra a mola, e levantado do corpo de prova, durante a movimentação do braço. Isto resolveu o problema.

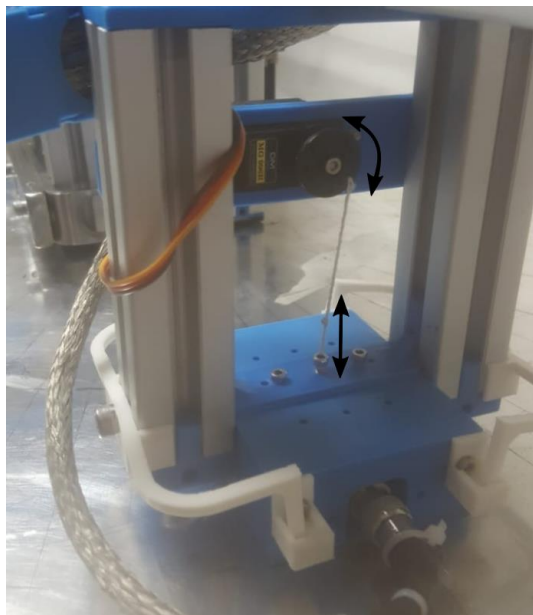


Figura 3-15 - Fotografia do sistema de movimentação vertical

c. Gerenciamento dos cabos

Os cabos de ambos motores e EMATs (central e periférico) precisam ser roteados de maneira conveniente, e que não atrapalhe a movimentação do manipulador. Mesmo os cabos cujos destinos estão

na estrutura central, estacionária, não permitem organização trivial. Já que não podem ser roteados diretamente para cima nem para fora, pois irão interferir com o braço ou estrutura periférica em algum ponto de sua trajetória circular. Para isso foram utilizadas '*dragchains*' feitas de peças impressas, com especificações adequadas. Cada link tem 30mm de largura, 45mm de comprimento, e 15mm de altura, com um ângulo máximo de rotação interno de 80°, alguns links podem dobrar externamente até 50°. A configuração final pode ser vista na Figura 3-16 abaixo.

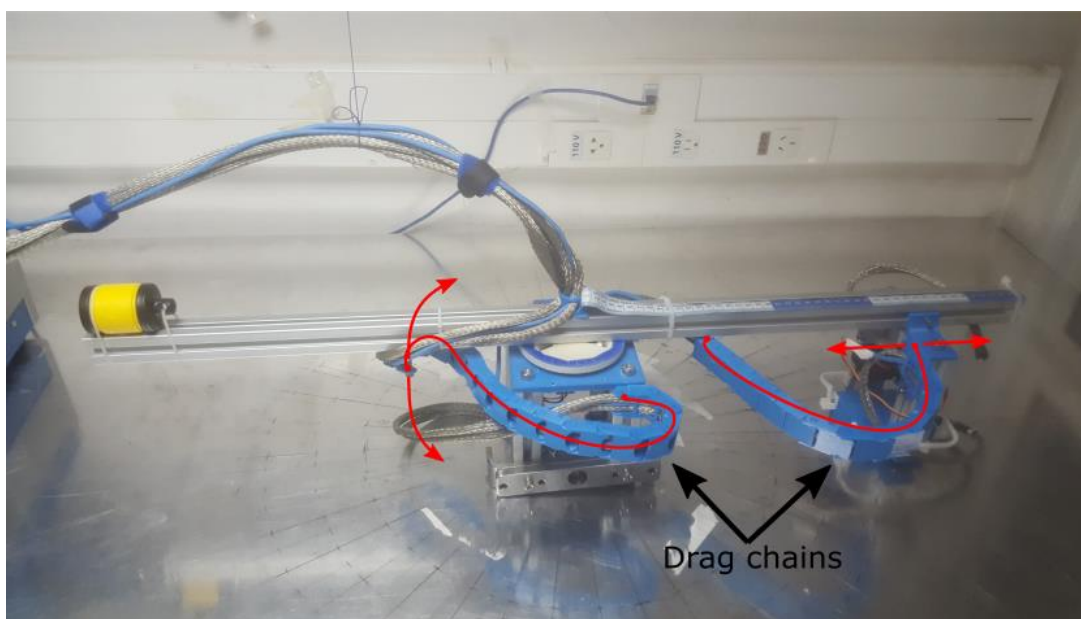


Figura 3-16 – *Dragchains* indicadas por setas pretas. Linhas vermelhas indicando o corpo da *dragchains*, Setas vermelhas indicando movimentação do ponto móvel. Posição aproximada de 0° e 350mm de raio.

Todos os cabos saem do manipulador acima do centro do braço e giram junto com o mesmo. A *dragchain* responsável por levar os cabos à estrutura periférica é relativamente simples, dobrando mais ou menos, apenas em uma direção, dependendo do raio utilizado na varredura. A *dragchain* que carrega os cabos para a estrutura central é mais complexa. Um de seus pontos de fixação gira em relação ao outro. Para possibilitar que o braço gire 360° ela dobra para os dois lados. Na Figura 3-17 podemos ver mais Fotografias em diferentes posições do braço. Vale comentar que a posição em 360° é distinta e oposta à origem no sentido de que a *dragchain* está enrolada para o lado oposto.

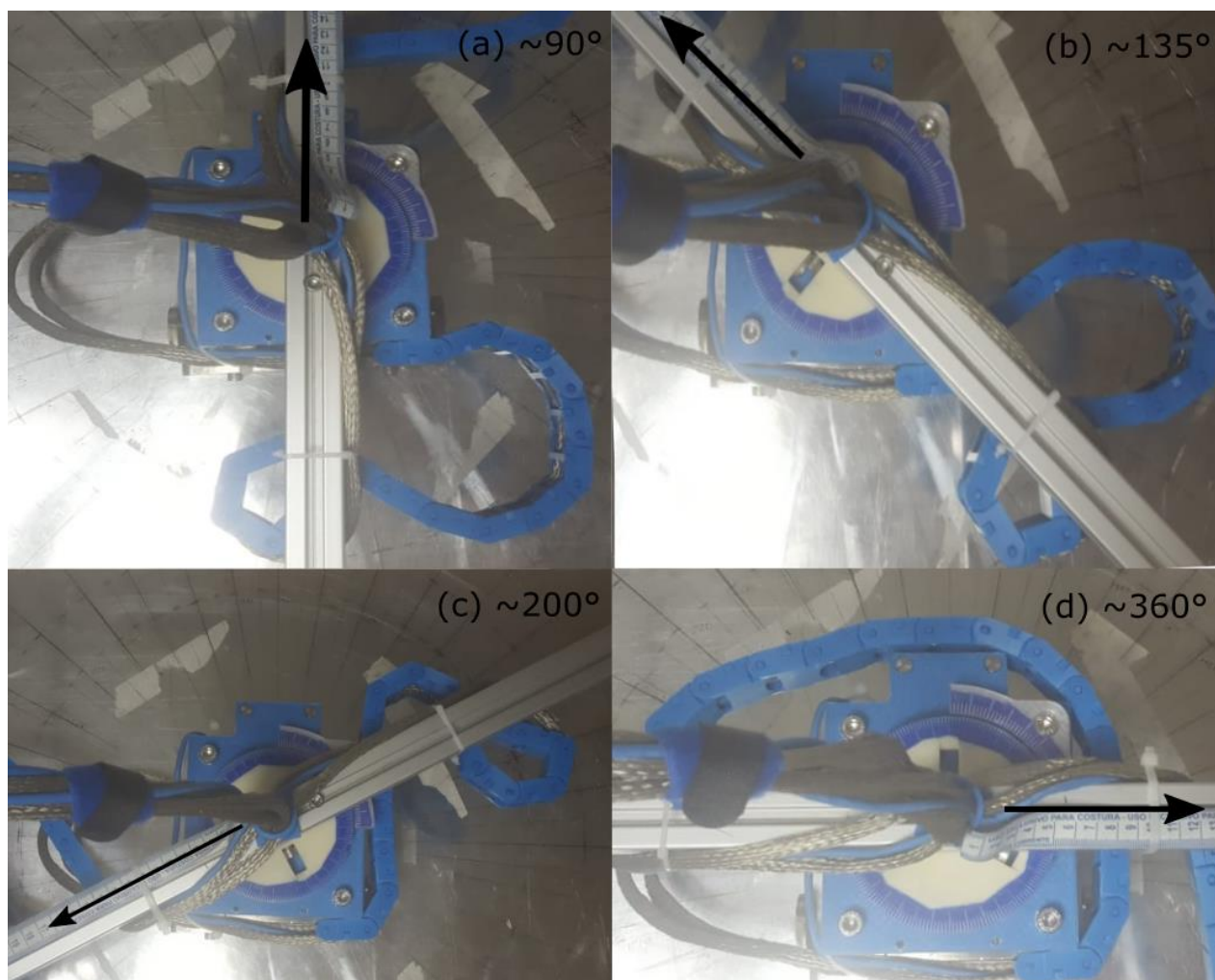


Figura 3-17 Algumas posições da *dragchain* central (a)~90°, (b)~135°, (c)~200°, (d)~360°.

As tolerâncias relacionadas aos cabos ficaram relativamente baixas. Originalmente as *dragchains* foram dimensionadas para apenas um cabo BNC, e não dois cabos com isolamento adicional ultimamente utilizados. O comprimento da *dragchain* central também foi aumentado para que ser presa do lado oposto ao original e o manipulador seguir a orientação padrão do círculo geométrico no espaço em que foi montado.

d. Jogo

Manipulando o braço é fácil identificar uma folga na junta central de modo que mesmo com o motor de passo travado o braço consegue girar um pouco. Essa folga é popularmente conhecida como 'jogo', termo que será utilizado doravante. Isto diminui a precisão e receptibilidade do manipulador. Mesmo que o menor passo teórico, seja 1/192 graus, a precisão real do braço é muito menor.

Esse jogo pode ser causado por alguns componentes e as tolerâncias em seus encaixes. Algumas causas possíveis são a junção do motor de passo com a caixa de redução, internamente a caixa, ou a junção entre a peça central do braço e a extrusão de alumínio do mesmo. Provavelmente o encaixe que é responsável pela maior parte do jogo é entre a peça central do braço e a redução. O mesmo encaixe que é mencionado extensivamente na subseção a desse capítulo.

Foi realizado um teste simples para comparar entre duas peças centrais do braço. A primeira, impressa em PETG e utilizada por duas semanas, a segunda impressa em ABS de alta temperatura nova. O ABS,

é uma material mais rígido, além disso o tempo de uso sob estresse possui um o efeito de 'creep' em peças impressas [24].

Nesse teste se mediu, aproximadamente o seno do ângulo com um relógio como mostra a Figura 3-18 abaixo. A medição, feita a 200mm do centro, consistiu em forçar o braço manualmente e anotar seu deslocamento. Na primeira peça foi aproximadamente 5.3mm, na segunda 2.0mm. Esses números correspondem a 1.52° e 0.57° respectivamente. Embora exista uma melhora significativa são necessários mais testes para atribuir corretamente a melhora à mudança de plástico ou ao fato de uma peça ser nova e a outra gasta, possivelmente sofrendo o efeito de 'creep'. Por hora, essa quantidade de jogo foi considerada aceitável.

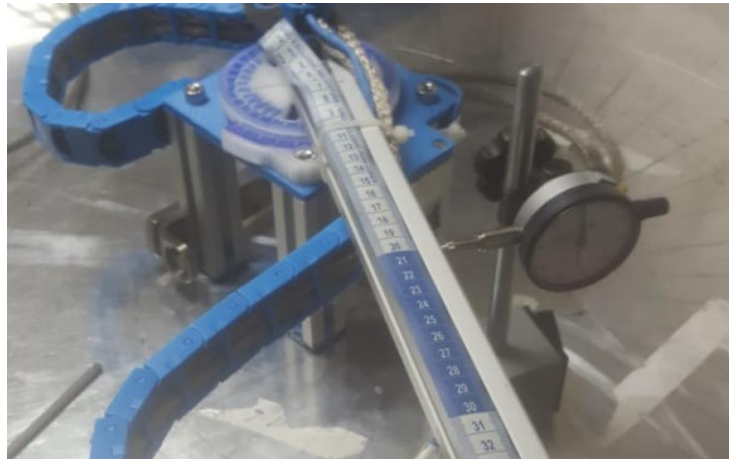


Figura 3-18 - Medição de folga da junta central

Os arquivos da montagem mecânica e das peças modeladas se encontram no apêndice C.

4. Eletrônica

Este capítulo trata do desenvolvimento da eletrônica necessária para o funcionamento do manipulador. A mesma deve ser organizada de maneira a facilitar o uso do manipulador aqui projetado e é composta dos seguintes componentes, ilustrados na Figura 4-1:

- O componente principal, o microcontrolador age como cérebro do manipulador executando os comandos recebidos. Foi escolhido o ESP32 devkit [25]. O mesmo possui dois núcleos de processamento com frequência de até 240 MHz, 512 kB de RAM, wifi e bluetooth integrados, e pinos suficientes para controlar todos os atuadores necessários.
- Para atuar o motor de passo NEMA 23 é necessário um driver externo. Utilizamos o TB6600 [26].
- O Servomotor 996R utilizado pode se conectar diretamente ao microcontrolador.
- São necessárias, também, fontes de alimentação de 5V para o microcontrolador e de 9-42V para o driver.

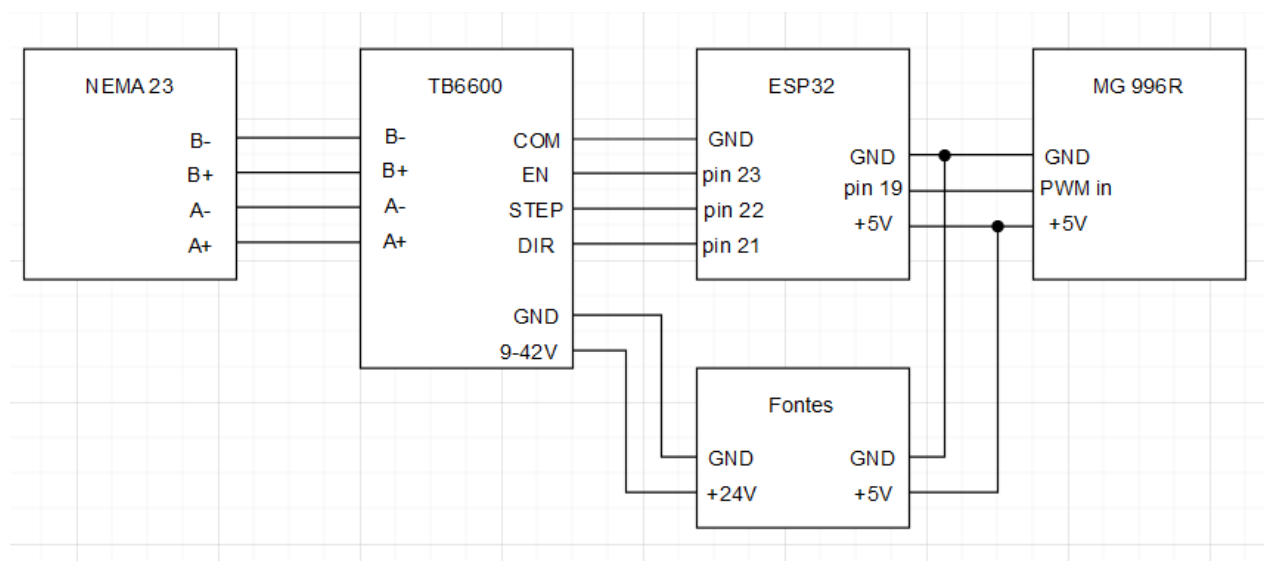


Figura 4-1 - Esquemático geral da eletrônica.

Para construir o bloco 'Fontes' há uma fonte chaveada de 24V, alimentada por tensão residencial padrão AC. E um módulo DCDC V7805-1000 [27]. Esse módulo possui entrada típica de 24V e saída de 5V com 1000mA máximo. Observa-se que o Servomotor apresenta um consumo de corrente máximo de 900mA (quando em movimentação) e o consumo máximo estimado do Microcontrolador é 141mA. Logo, teoricamente seria necessário uma fonte de 5V capaz de fornecer maior corrente. Contudo, o consumo do conjunto foi medido, nunca passando de 600mA.

O bloco Fontes, portanto ficou como indicado na Figura 4-2 abaixo.

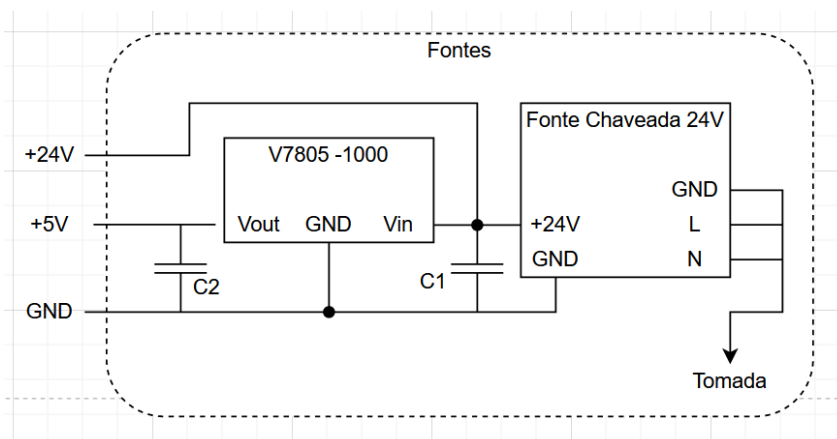


Figura 4-2 - Esquemático do bloco de fontes.

Para ajudar na montagem do circuito foi utilizado uma placa universal. Nela foram montados os componentes de menor tamanho, incluindo o microcontrolador, além de conectores do tipo bornes para facilitar a montagem do circuito completo. A Figura 4-3 apresenta uma fotografia dessa placa, com as conexões marcadas.

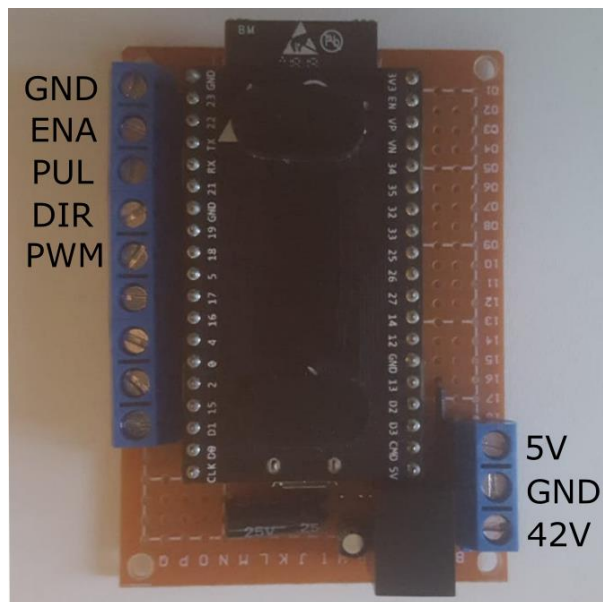


Figura 4-3 - Fotografia da placa do microcontrolador com marcação nas conexões necessárias.

Depois de um teste completo da eletrônica foi modelado e impresso uma caixa para proteger os componentes. Abaixo podemos ver o modelo assim como uma fotografia da montagem final na Figura 4-4.

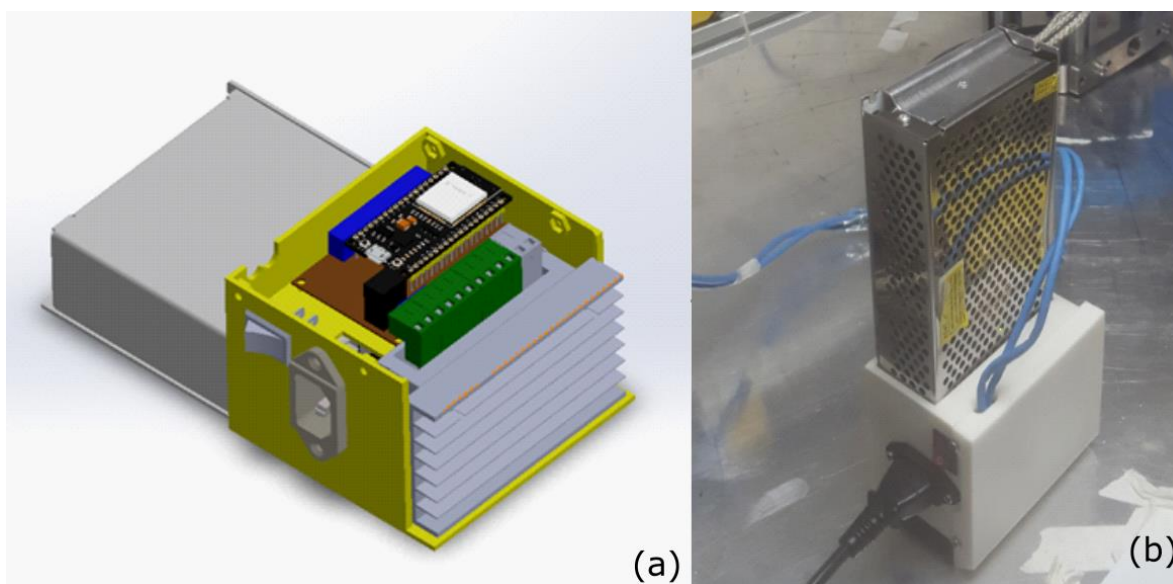


Figura 4-4 - Montagem da caixa da eletrônica (a) Modelo, sem tampa. (b) Fotografia da montagem final.

Os arquivos da montagem eletrônica das peças modeladas se encontram no apêndice C.

5. Software

O software necessário para o projeto é dividido em duas partes. O script é executado no computador, o qual orquestra a varredura, e o firmware que executado no microcontrolador do braço. A comunicação entre eles é essencial.

a. Firmware - Arduino

O firmware, escrito utilizando a plataforma do Arduino, também pode ser dividido em duas partes:

Comunicação via wifi:

O microcontrolador age como um ponto de acesso, ao qual o computador se conecta via uma placa de rede externa. O firmware utiliza uma biblioteca de Web Server [28] que faz todo o gerenciamento de rede. As URLs (Localizador de Recursos Uniforme) que o computador requisita consistem de um endereço ip, um comando e, em alguns casos, parâmetros adicionais. Utilizando a biblioteca o comando de cada URL é interpretado e atribuído sua função correspondente do firmware. Um exemplo que poderia ser requisitada ao manipulador seria <http://192.168.4.1/stop>. Ela contém o ip que foi utilizado (192.168.4.1), uma barra (/) e o comando que requisita parrar o movimento do braço. Segue abaixo a lista dos principais comandos atualmente implementados, seus parâmetros possíveis e respostas. Essa lista também está disponível na home do servidor, página sem nenhum comando, apenas o ip.

- `.../move?=valueX&=valueY`
Comando para movimento relativo onde valueX representa o número de passos do eixo X, um número inteiro, e o eixo andar value X/192 graus.
Retornará "ok" e executará o movimento ou "wait" caso o manipulador já esteja em movimento. Com o retorno "wait" o comando não será executado.
- `.../moveTo?=valueX&=valueY&=valueZ&=valueA&=valueC`
Comando para movimento relativo com mesma estrutura do comando relativo.
- `.../trig`
Sobe a tensão do pino trigger por um pequeno intervalo. Sempre retorna "ok".
- `.../set_MaxSpeed?=value`
Configura a velocidade máxima de todos os eixos para value em steps/sec
Sempre retorna "ok"
- `.../set_MaxAccel?=value`
Configura a velocidade máxima de todos os eixos para value em steps/sec²
Sempre retorna "ok"
- `.../get_pos`
Retornará "x=value1 y=value2" correspondente à posição atual de cada eixo
- `.../stop`
Desacelera ao máximo o braço até parar. Sempre retorna "ok"
- `.../target_pos`
Retornará "x=value1 y=value2 z=....." Valores referentes à posição alvo de cada eixo.
- `.../set_currentPos?=valueX&=valueY`
Troca a referência da posição atual do braço para (valueX,valueY) Caso não sejam passados argumentos será configurado a posição atual como origem .Sempre retornará "ok"

Caso algum argumento seja omitido o valor 0 é considerado como padrão.

Como é esperado que os pedidos URL sejam feitos por outros programas via linha de comando não foi implementada nenhuma interface nas páginas do servidor.

Pela maneira como o programa foi feito somente é possível passar um comando por vez. Não há um buffer de comandos, nem de movimentos. Se um pedido de movimento for feito enquanto a máquina está se mexendo, o mesmo é ignorado, e o servidor responde "wait" em vez de "ok". Isso é consequência da maneira como foi implementado o movimento utilizando a biblioteca AccelStepper [29]. Além disso, qualquer comando de movimento também age sobre o servo motor, garantindo que o manipulador só se mova com o EMAT receptor levantado. Ao chegar nas coordenadas desejadas o manipulador desce o EMAT e depois considera o movimento finalizado.

Controle dos drivers dos motores de passo:

Embora o hardware mecânico atual apenas utilize um eixo de movimentação automatizado, como foi prevista a motorização do eixo radial, se manteve o controle de dois eixos em software, aproveitando código já feito em projetos anteriores.

Apesar da biblioteca possuir uma maneira de sincronizar vários eixos, essa funcionalidade não foi utilizada pois não suporta controle de aceleração, apenas velocidades constantes. Utilizamos então eixos separados, na visão da biblioteca, em conjunto com um código para sincronizá-los.

Esse código de sincronização se baseia no fato de que se utilizarmos uma aceleração e velocidade máxima proporcional entre eixos, a distância percorrida para qualquer intervalo de tempo apresenta essa mesma proporcionalidade. Esse fato é fácil de visualizar se traçarmos curvas de velocidade x tempo para os eixos, com a distância percorrida sendo a área abaixo da curva, Figura 5-1. Esse método deixa os eixos no máximo um passo fora de sincronia.

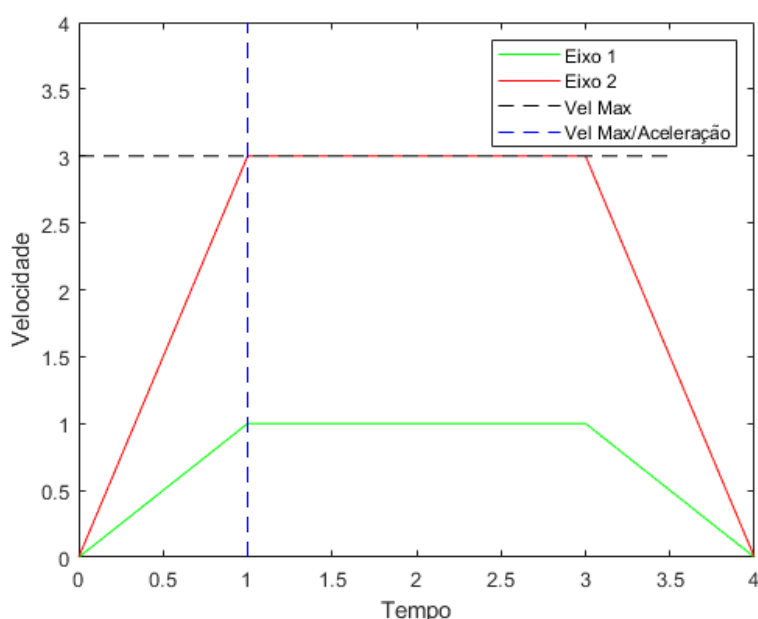


Figura 5-1 - Curva de velocidade de dois eixos.

Vale comentar que o firmware, faz uso de ambos os núcleos de processamento de acordo com a divisão mencionada acima. Um núcleo trata respostas da rede, e o outro realiza os passos necessários. Isso discrimina as funções de rede que possuem um tempo de execução muito maior. Dessa forma o *timing* dos passos não é comprometido dado que a biblioteca AccelStepper funciona à base de *polling* e não de interrupção.

O firmware encontram-se no apêndice A.

b. Matlab® Script

Scripts de Matlab® são utilizados no computador para controlar a varredura. Foi construída uma função de Matlab® que utiliza o comando `urlread` para se comunicar com o manipulador utilizando o protocolo adequado. Essa função `'cnc_moveTo'` requisita o movimento desejado, e confere se ele foi executado, apenas retornando quando o manipulador já estiver movido e parado novamente.

A função pode ser utilizada em outro script mais complexo que realiza a varredura, iterando sobre posições diferentes. Esse script pode também se comunicar com o osciloscópio e é responsável por armazenar os dados em um arquivo `'.mat'`.

Os scripts de Matlab® encontram-se no apêndice B.

6. Resultados

Após a construção do projeto aqui concebido, foram realizadas varreduras com o objetivo de comparar resultados obtidos por varredura manual e utilizando o manipulador. Todas utilizaram os mesmos transdutores e equipamentos e configurações listadas abaixo. A única distinção foi o uso do manipulador, e a resolução da varredura.

- EMAT receptor: Sonemat Ltd. SHD2531S de 25mm de comprimento de onda nominal, posicionado a 320mm de raio.
- EMAT emissor: Construído no laboratório. Enrolamento é feito com uma placa de circuito impressa (do inglês PCB) e os imãs de (20x10x5mm) estão em uma caixa impressa. O comprimento de onda nominal é de 22mm. Figura 6-1
- Corpo de prova: Chapa de alumínio considerada uniforme e extensa com 1.5mm de espessura.
- Osciloscópio: Tektronix DPO 4054. Escala vertical: 100mV; Escala Horizontal: 40us; Average 512.
- Gerador de função (pulso), Amplificador de potência e Amplificador de sinal com suas funções todas realizadas pelo mesmo equipamento: Ritec® RPR 4000 pulse receiver configurado da seguinte maneira: Frequência: 141.5kHz; ciclos: 8; duração de pulso: 56.5 μ s; Control (ganho de saída): 0; tensão de saída: 40V; Ganho de recepção: 72.8dB; Filtro passa baixa: 800kHz; Filtro passa alta: 50kHz. Rf output conectado ao EMAT emissor; Input ao EMAT receptor. Trigger e output para canais do osciloscópio.

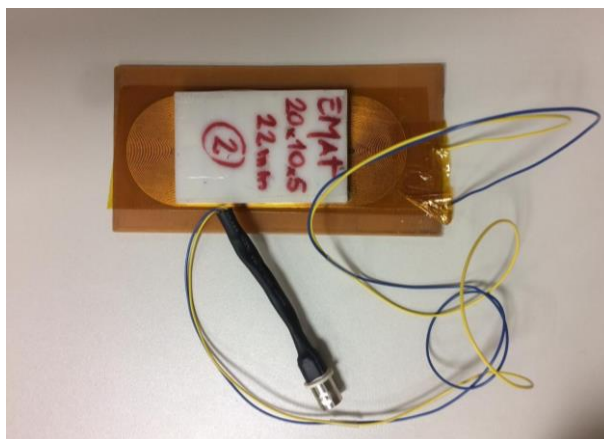


Figura 6-1 - Fotografia EMAT construído no laboratório 22mm.

A primeira varredura foi realizada à mão com uma resolução de 10° (37 passos), cobrindo toda a circunferência incluindo os pontos equivalentes 0° e 360° , ou seja, a posição angular inicial de varredura é percorrida duas vezes para fins de permitir verificar se não houve algum erro acumulado ao longo da varredura, motivo pelo qual temos 37 passos e não 36. A mesma foi realizada por duas pessoas e demorou aproximadamente 16 minutos. Nessa varredura, uma pessoa operou o computador, conferindo as medições e confirmando a passagem para o próximo ponto, enquanto outra moveu o EMAT para as posições necessárias. Caso a mesma fosse realizada por um operador sozinho teria sido gasto consideravelmente mais tempo, pois a pessoa teria de realizar as duas tarefas intermitentemente, tendo que se deslocar da posição do computador até a posição do EMAT receptor em cada iteração. Esse tempo não inclui nenhuma preparação anterior ou posterior.

Posteriormente foram realizadas duas varreduras, operados por uma única pessoa, utilizando o manipulador. Elas tiveram resolução de 10° e 2° (181 passos) e gastaram aproximadamente 9 e 37 minutos respectivamente. O número de passos é dado pela mesma lógica da varredura anterior. Principalmente utilizando passos abaixo de 15° , o tempo de movimentação angular é consideravelmente menor do que a movimentação vertical e o resto do tempo necessário para realizar a medida. Portanto podemos considerar o tempo gasto proporcional ao número de medidas (passos) realizadas. Com isso, calculamos aproximadamente 26 segundos por passo na varredura manual, 52 considerando o tempo de dois operadores. Na varredura de 10° , que representa uma melhor comparação, temos 15 segundos por passo. Isso caracteriza um gasto aproximadamente 3.5 vezes menor de tempo.

Vale ainda comentar também, que o script de Matlab® utilizado para realizar a varredura necessita de input do operador a cada leitura. Para realizar cada medida o operador deve confirmar que a mesma foi realizada corretamente, ou escolher repeti-la. Devido a isso, mesmo utilizando o manipulador é

necessário a presença de um operador durante toda a varredura. Caso o script fosse alterado para aceitar e passar automaticamente para o próximo ponto, seria possível tornar a varredura ainda mais rápida, e sem nenhum input necessário.

Depois de concluído o procedimento, é obtido um sinal tipo forma de onda, similar ao apresentado na Figura 6-2, para cada ponto no qual uma medida foi realizada. Podemos observar duas regiões bem definidas no Ascan de 0° . A primeira (de 0 a aproximadamente $75\mu s$) corresponde à interferência de alta tensão de excitação. A segunda (de 100 a aproximadamente $175\mu s$) corresponde ao sinal da onda captada pelo EMAT receptor. O tempo de chegada da onda depende da velocidade da mesma, dado que a distância é conhecida, pela definição de velocidade podemos relaciona-la com o tempo gasto para a onda percorrer o caminho entre transdutores. Nesse caso a velocidade teórica da onda cisalhante no alumínio de $3111m/s$ resultando em um tempo de $102.86\mu s$ para percorrer os $320mm$, esse tempo é bastante consistente com o observado no Ascan. Além disso, a amplitude e forma da onda variam de acordo com o ângulo no qual a medida foi feita, como pode ser observado nos diferentes sinais da Figura 6-2, correspondentes a várias posições angulares.

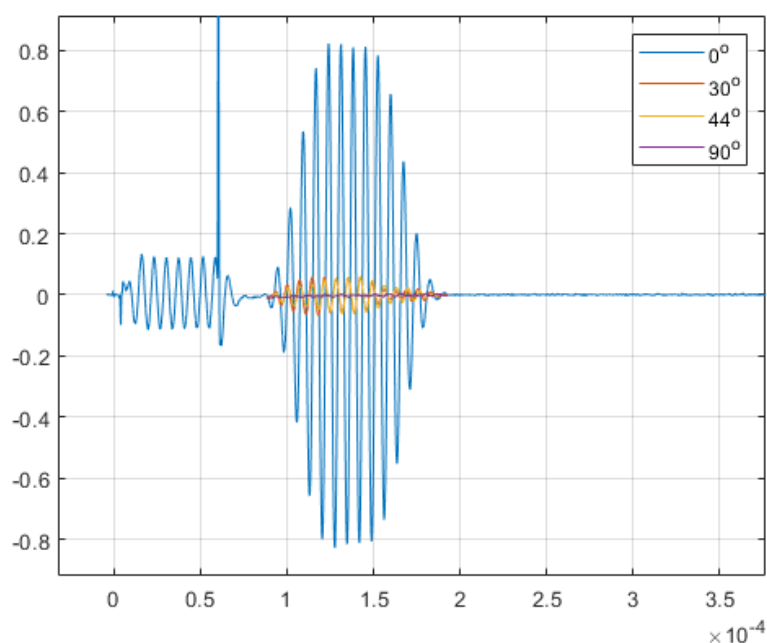


Figura 6-2 - Formato de um Ascan em diferentes posições angulares. Linha azul correspondente a 0° , vermelha a 30° , amarela a 44° e roxa a 90° .

Para analisar a qualidade da varredura utilizamos os dados obtidos para gerar um diagrama de radiação. Esse diagrama é feito desenhando-se um ponto, em coordenadas polares, para cada medição realizada. O valor de pico a pico normalizado, da janela correspondente a onda recebida, nos dá o raio do ponto, nesse caso em dB. O ângulo do diagrama é o próprio ângulo do EMAT no qual a medida foi feita. Na Figura 6-3 abaixo temos os diagramas de duas varreduras realizadas. O gráfico resultante deve ter amplitude máxima nas direções principais, ou seja, em 0° e 180° , onde o gráfico deveria ser 0dB. Além disso, os diagramas de radiação também devem ser simétricos, sob o eixo vertical e horizontal.

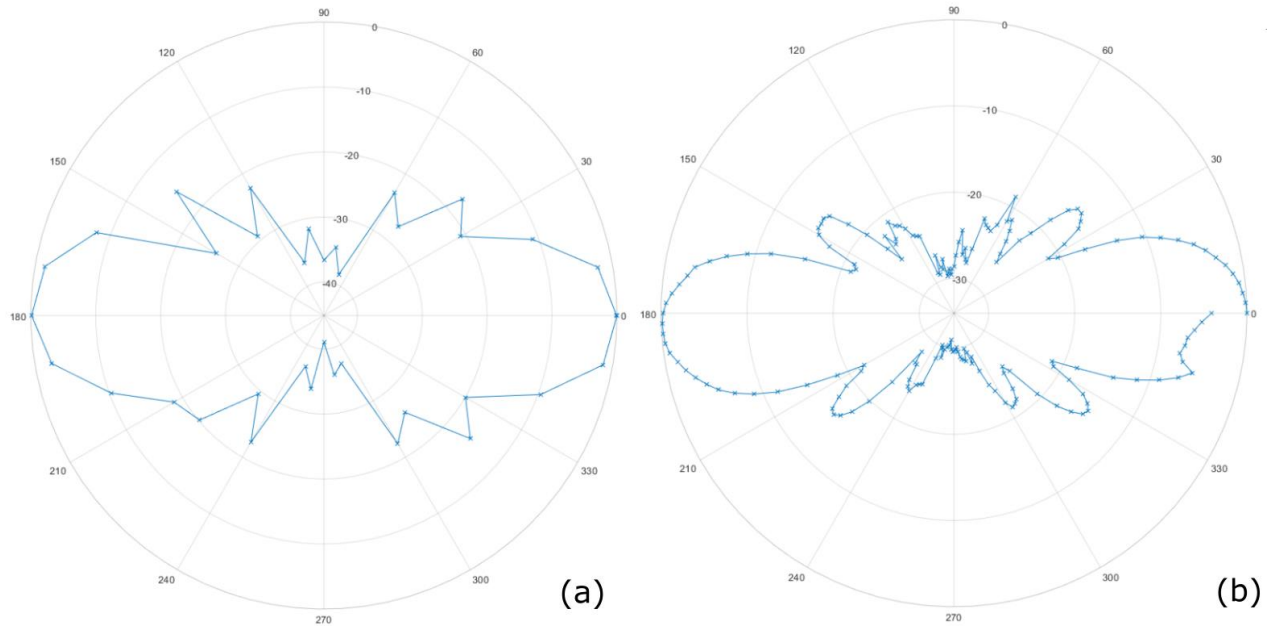


Figura 6-3 - Diagrama de radiação (a) Varredura manual, passo 10° (b) Varredura automatizada, passo 2°.

Comparando os gráficos da figura acima podemos observar a vantagem que o número maior de pontos nos dá. Podemos observar, que as varreduras feitas com o braço mecânico possuem uma região de 345° a 360° na qual as medidas ficaram bem inconsistentes. A provável causa é uma o mal contato entre a caixa do EMAT receptor e o corpo de prova, diminuindo a intensidade da medida realizada. Isso pode ser causado, por exemplo, por alguma sujeira que impeça esse contato.

7. Conclusões e recomendações futuras

Com os resultados obtidos podemos concluir que o projeto foi bem-sucedido em seus objetivos. A realização das varreduras automatizadas requer menos esforço e menos de um terço do tempo, comparadas às feitas manualmente. Isso possibilita a aquisição de mais dados e ajuda o desenvolvimento dessas tecnologias. O manipulador é uma ferramenta útil para o laboratório e para a fabricação e caracterização de transdutores EMATs.

Nesse capítulo também iremos apresentar as principais recomendações futuras para o projeto. Salienta-se, contudo, o manipulador ainda não foi utilizado extensivamente. É muito provável que surgirão novas recomendações feitas por seus operadores a medida que utilizam o equipamento.

a. Alterar maneira de fixar a caixa do EMAT receptor

A caixa do EMAT receptor atualmente é montada no resto da estrutura periférica por via de 2 parafusos mostrados na Figura 7-1. A caixa do EMAT e a peça ao qual é presa foram modeladas antes da inclusão do sistema de movimentação vertical no manipulador. O furo central que continha um parafuso agora é utilizado para passar a corda que prende ao servo motor.

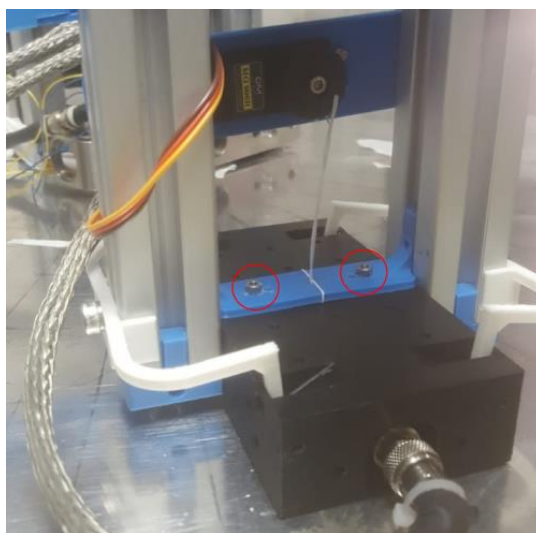


Figura 7-1 - Fotografia do sistema de fixação da caixa do EMAT receptor. Parafusos circundados em vermelho.

Esse sistema não é prático, tendo que apertar parafusos com pouco acesso. Dado que os EMATs são peças que se espera trocar regularmente esse sistema de fixação deve ser alterado de modo que fique mais prático e fácil. Essa alteração não deve ser complexa de ser modelada, necessitando apenas a alteração da peça ressaltada em vermelho na Figura 7-2.

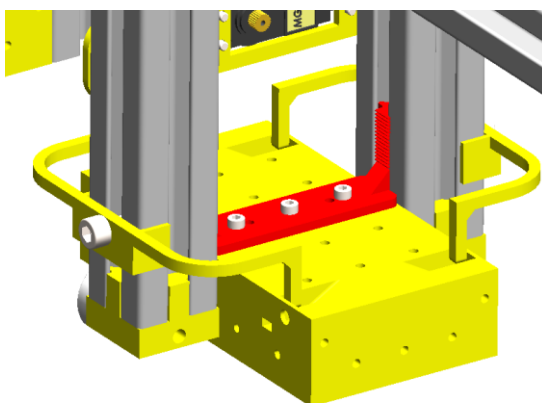


Figura 7-2 - Modelo da peça de montagem do EMAT periférico ressaltada vem vermelho.

A nova peça poderia ser aparafusada a caixa do EMAT pela lateral, o que facilitaria o acesso. Ela também poderia utilizar porcas T específicas para deslizar verticalmente nos perfis de alumínio. Isso tornaria o movimento mais suave que o encaixe impresso atual. Além disso, seria benéfico ter uma maneira fácil de regular a altura da corda. O novo design poderia incorporar essa regulagem.

b. Movimentação vertical e servomotor

A adição do servomotor permitiu a movimentação vertical do EMAT receptor e solucionou o problema de atrito do mesmo. Contudo, esse sistema opera com folgas muito pequenas, em vários quesitos. A regulagem de altura é difícil, mesmo utilizando todo o alcance de movimento do servomotor ariscamos não levantar o EMAT suficientemente em algumas partes da trajetória. Poderíamos alterar o raio da polia, aumentando a movimentação. Contudo seria necessário um estudo mais profundo que compare o torque do motor com o peso do EMAT e, criticamente, tensão nas molas impressas. Além disso o servomotor também opera sem folga teórica de potência elétrica como mencionado anteriormente. A maneira como a corda está presa também deve ser alterada para diminuir o atrito e desgaste.

Considerando isso, trocar o servomotor por um motor de passo ofereceria uma solução mais robusta. A regulagem de força, e distância vertical seria mais fácil de realizar. Além disso, os motores e drivers necessários já estão disponíveis.

c. Motorização do eixo radial.

Essa alteração é mais complexa pois depende da mecânica, eletrônica e software. Contudo, como ela já foi prevista desde o início no projeto, nenhuma adaptação grande deve ser necessária.

No que concerne a montagem mecânica, e inclusão de um motor de passo, com barra roscada é simples, ambos ficariam na parte superior do braço, presos na extrusão de alumínio. A peça da estrutura periférica que prende no braço seria trocada por uma que seria fixa na castanha. A organização de cabos seria trivial, não necessitando de mais nenhuma *dragchain*. Os modelos iniciais foram feitos, mas ainda não testados (Figura 7-3).

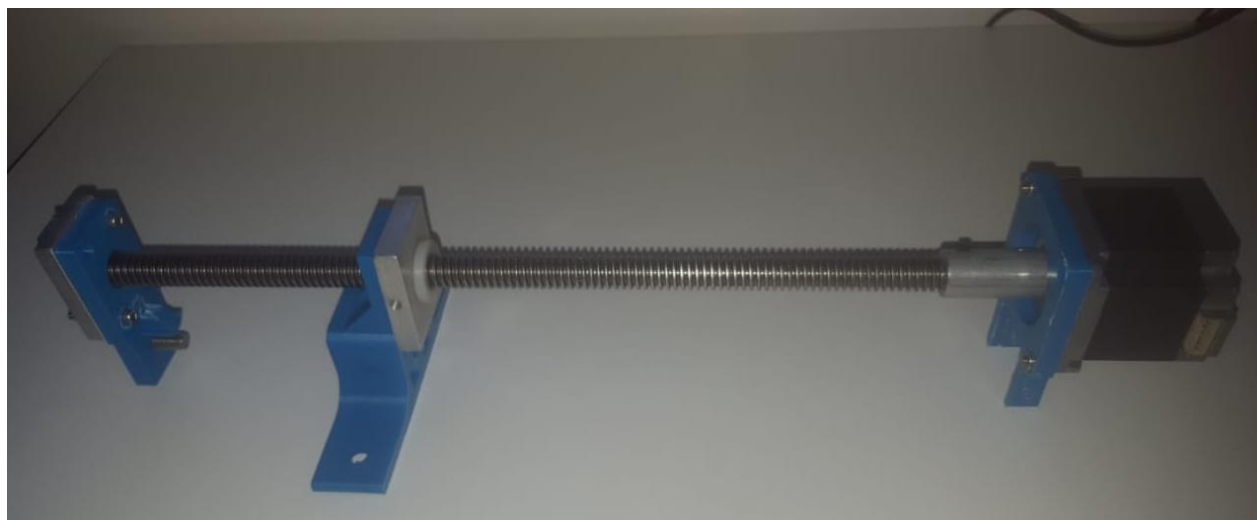


Figura 7-3 - Fotografia do teste isolado do mecanismo de movimentação axial.

No que concerne à parte eletrônica seria necessário adicionar mais um driver do motor de passo. Os pinos de conexão com o microcontrolador já estão designados no firmware. A montagem atual já possui todos os conectores e potência necessária. Apenas seria necessário adaptar a modelagem da caixa da eletrônica para o espaço de um segundo driver.

O firmware não precisa de nenhuma alteração. Já o script de Matlab® 'CNC_moveTo' precisa ser alterado para receber o segundo parâmetro pertinente a posição radial. Ele também precisa incluir a constante de passos por distância específica do hardware.

d. Melhorar o jogo do eixo central.

Houve uma grande melhora dessa folga a cada iteração da peça central do braço. Contudo, é necessário um estudo ainda mais compreensivo. Não somente dessa peça em questão, mas também capaz de identificar todas as causas desse jogo. É possível que o redesenho do encaixe, de maneira menos limitada em altura resolva a situação. A fabricação de uma peça metálica também pode ser uma solução viável. Pode ser explorada inclusive a utilização de outra redução com um encaixe em chaveta. Esse encaixe é vital para a precisão das medidas realizadas e o objetivo final do projeto inteiro.

8. Referências

- [1] L. W. J. Schmerr, *Fundamentals of Ultrasonic Nondestructive Evaluation, A Modeling Approach*, Second Edition, Ames, Springer, 2016.
- [2] J. L. Santin, *Ultra-som, técnica e Aplicação*, Rio de Janeiro: Qualitymark, 1997.
- [3] J. D. N. Cheeke, *Fundamentals and Applications of Ultrasonic Waves*, 2nd edition, Boca Raton: CRC Press, 2017.
- [4] J. L. Rose, *Ultrasonic Guided Waves in Solid Media*, Cambridge: Cambridge Univ. Press, 2014.
- [5] H. O. M. Hirao, *EMATs for Science and Industry: Noncontacting Ultrasonic Measurements*, New York: Springer, 2003.
- [6] R. Baiotto, *Desenvolvimento e Caracterização de Transdutor Acústico Eletromagnético de Imãs Permanentes Periódicos*, Dissertação de Mestrado, Porto Alegre: Universidade Federal do Rio Grande do Sul, 2014.
- [7] H. Miao e F. Li, "Shear horizontal wave transducers for structural health monitoring and nondestructive testing: A review", In: *Ultrasonics*, Volume 114, July 2021, 106355
- [8] P. Khalili e F. Cegla, "Excitation of Single-Mode Shear-Horizontal Guided Waves and Evaluation of Their Sensitivity to Very Shallow Crack-Like Defects", in *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, Volume 68, no. 3, pp. 818-, 2021.
- [9] C. Camerini, C. Marinho, A. Franzoi, S. Ibajé, M. Freitas e J. Weid, "Feeler pig – a new tool for multi-size pipeline inspection", In: *European non-destructive testing standards conference*, Berlin, Conference 9, 2006.
- [10] P. A. Petcher e S. Dixon, "Mode mixing in shear horizontal ultrasonic guided waves"; In *Nondestructive Testing and Evaluation*, Volume 32, 2017; pages 113-132; [Online] disponível em <http://dx.doi.org/10.1080/10589759.2016.1184268>, 2017.
- [11] S. Hill, S. Dixon, S. H. Reddy, P. Rajagopal e K. Balasubramaniam, "A New Electromagnetic Acoustic Transducer Design for Generating Torsional Guided Wave Modes for Pipe Inspections", In: *AIP Conference Proceedings*, 2017. [Online] Disponível em: <https://aip.scitation.org/doi/abs/10.1063/1.4974597>
- [12] O. Trushkevych, M. Tabatabaeipour, S. Dixon, M. Potter, et al "Miniaturised SH EMATs for fast robotic screening of wall thinning in steel plates", In: *IEEE Sensors Journal*, vol. 21, no. 2, pp. 1386-1394, 2021.
- [13] M. Clough, M. Fleming e S. Dixon, "Circumferential guided wave EMAT system for pipeline screening using shear horizontal ultrasound", In: *NDT & E International*, Volume 86, March 2017, Pages 20-27.
- [14] N. Suresh e K. Balasubramaniam, "Quantifying the lowest remnant thickness using a novel broadband wavelength and frequency EMAT utilizing the cut-off property of guided waves", In: *NDT & E International*, Volume 116, December 2020, 102313.
- [15] C. Hakoda, V. K. Chillara e C. Pantea, "The effect of a transducer's spatial averaging on an elastodynamic guided wave's wavenumber spectrum", Los Alamos: Los Alamos National Laboratory, 2021.
- [16] L. M. Martinho, *Enhancement of the sensitivity to deformation of broadband ultrasonic signals in plates using spectral phase selectivity*, Projeto final de graduação, Rio de Janeiro: Pontifícia Universidade Católica, 2020.

- [17] Motor de passo NEMA 23 23KM-C Data Sheet; [Online] Disponível em: http://cnc25.free.fr/documentation/moteurs%20pap/pap_nmb.pdf; último acesso em 09/06/2021.
- [18] V. B. Bhandari, *Design of Machine Elements*; 3rd edition; New York: Tata McGraw-Hill Education, 1994.
- [19] S. Huang, W. Zhao, Y. Chang e S. Wang, "Study on the lift-off effect of EMAT", In: *Sensors and Actuators A: Physical*, Volume 153, Issue 2, 3 August 2009, Pages 218-221, 2009.
- [20] S. Saraubh, S. Kumar, K. K. Jain, S. K. Behera, D. Gandhi, S. Raghavendra e K. Kalita "Design of Suspension System for Formula Student Race Car" In: *Procedia Engineering*, Volume 144, 2016, Pages 1138-1149, 12th International Conference on Vibration Problems, 2016.
- [21] M. F. Mwema e E. T. Akinlabi, "Basics of Fused Deposition Modelling (FDM)". In: *Fused Deposition Modeling. SpringerBriefs in Applied Sciences and Technology*. Springer, Cham. Disponível em: https://doi.org/10.1007/978-3-030-48259-6_1, 2020.
- [22] A. R. Torrado e D. A. Roberson, "Failure Analysis and Anisotropy Evaluation of 3D Printed Tensile Test Specimens of Different Geometries and Print Raster Patterns", In: *Journal of Failure Analysis and Prevention*, 2016.
- [23] Servo motor MG996R - DataSheet; [Online] Disponível em: https://components101.com/asset/sites/default/files/component_datasheet/MG996R-Datasheet.pdf, último acesso em 01/06/2021.
- [24] T. Tezel e V. Kovan, *Effects of the printing parameters on short-term creep behaviors of three-dimensional printed polymers*; *J Appl Polym Sci*, 136, 47564. [Online] Disponível em: <https://doi.org/10.1002/app.47564>, 2019.
- [25] ESP 32 board, [Online] Disponível em: <https://docs.espressif.com/projects/esp-idf/en/v3.1.5/get-started/get-started-devkitc.html>, último acesso em 31/05/2021.
- [26] Stepper driver Tb6600 Data Sheet, [Online] Disponível em: https://www.mcielectronics.cl/website_MCI/static/documents/TB6600_data_sheet.pdf, último acesso em 02/06/2021.
- [27] 5V Switching Regulator Data Sheet, [Online] Disponível em: <https://www.cui.com/product/resource/v78xx-1000.pdf>, último acesso em 03/06/2021.
- [28] Biblioteca WebServer ESP32, [Online] Disponível em: <https://github.com/espressif/arduino-esp32/blob/master/libraries/WebServer/src/WebServer.h>, último acesso em 30/05/2021.
- [29] Biblioteca AccelStepper, [Online] Disponível em: <http://www.airspayce.com/mikem/arduino/AccelStepper/>, último acesso em 02/06/2021.

Apêndice A

Firmware

Nessa subseção é apresentado o código utilizado no firmware. O arquivo 'ESP_CNC_2_dualCoreNewPins.ino' escrito em C++, na Arduino IDE. Também é apresentado o arquivo complementar 'contents.h'. Esse código é compilado e enviado e salvo no microcontrolador, que o executa durante o funcionamento do braço.

ESP_CNC_2_dualCoreNewPins.ino:

```
#include <WiFiClient.h>
#include <WiFi.h>
#include <WebServer.h>
#include <ESP32Servo.h>
#include <AccelStepper.h>
#include "contents.h"

TaskHandle_t Task1;

const int STOP_X = 18;
const int STOP_Y = 33; //

const int EN_X = 23;
const int STEP_X = 22;
const int DIR_X = 21;

const int EN_Y = 17; //
const int DIR_Y = 18; //
const int STEP_Y = 5; //

const int TRIG = 13; //

int servoPin = 19;
int servoUP = 179;
int servoDOWN = 1;
int servoPos = servoUP;

bool alwaysOn = true;
bool servoOn = true;
bool homing = false;
int trig_on = 0;
long biggestlong = 2147483647;
int Default_trig_speed = 100;
int f_ready = 1;

int MAXSPEED = 2000; // in steps/sec
int MAXACCEL = 3000; // in steps/sec²

// Replace with your network credentials
char* ssid = "cnc";
char* password = "pigpig1234";

IPAddress ip(192, 168, 0, 1); //
IPAddress netmask(255, 255, 255, 0);
WebServer server(80); //instantiate server at port 80 (http port) //2226 for esp32

AccelStepper stepper1(AccelStepper::DRIVER, STEP_X, DIR_X);
AccelStepper stepper2(AccelStepper::DRIVER, STEP_Y, DIR_Y);

AccelStepper trigger(AccelStepper::DRIVER, TRIG);

Servo myservo; // create servo object to control a servo
```

```
//int servoPos = 0; // variable to store the servo position
// Recommended PWM GPIO pins on the ESP32 include 2,4,12-19,21-23,25-27,32-33
```

```
void prepareRelativeMove(int positions[5]) {
    if (servoOn) {
        servoPos = servoUP;
        //myservo.write(179); // tell servo to go to position in variable 'pos'
        delay(500);
    }
    if (positions[0]) {
        digitalWrite(EN_X, LOW);
    }
    if (positions[1]) {
        digitalWrite(EN_Y, LOW);
    }
    float proporcao[5] = {1, 1, 1, 1, 1};
    int maiorDistancia = 0;
    for (int i = 0; i < 5; i++) {
        if (maiorDistancia < abs(positions[i])) {
            maiorDistancia = abs(positions[i]);
        }
    }
    if (!maiorDistancia) {
        return;
    }
    for (int i = 0; i < 5; i++) {
        proporcao[i] = ((float)abs(positions[i]) / (float)abs(maiorDistancia));
    }
    stepper1.setMaxSpeed((proporcao[0]*MAXSPEED));
    stepper2.setMaxSpeed((proporcao[1]*MAXSPEED));

    stepper1.setAcceleration(proporcao[0]*MAXACCEL);
    stepper2.setAcceleration(proporcao[1]*MAXACCEL);

    stepper1.move(positions[0]);
    stepper2.move(positions[1]);

    f_ready = 0;
}

void prepareAbsoluteMove(int positions[5]) {
    if (servoOn) {
        servoPos = servoUP;
        //myservo.write(179); // tell servo to go to position in variable 'pos'
        delay(500);
    }
    //usar o codigo moveTo
    float proporcao[5] = {1, 1, 1, 1, 1};
    int maiorDistancia = 0;
    stepper1.moveTo(positions[0]);
    stepper2.moveTo(positions[1]);
    //transformar o vetor de positions em relativo
    positions[0] = stepper1.distanceToGo();
    positions[1] = stepper2.distanceToGo();

    //codigo igual ao relativo
    for (int i = 0; i < 5; i++) {
        if (maiorDistancia < abs(positions[i])) {
            maiorDistancia = abs(positions[i]);
        }
    }
    if (!maiorDistancia) {
        return;
    }
}
```

```

    for (int i = 0; i < 5; i++) {
        proporcao[i] = ((float)abs(positions[i]) / (float)abs(maiorDistancia));

        //Serial.println(proporcao[i]);
    }
    stepper1.setMaxSpeed((proporcao[0]*MAXSPEED));
    stepper2.setMaxSpeed((proporcao[1]*MAXSPEED));

    stepper1.setAcceleration(proporcao[0]*MAXACCEL);
    stepper2.setAcceleration(proporcao[1]*MAXACCEL);

    if (positions[0]) {
        digitalWrite(EN_X, LOW);
    }
    if (positions[1]) {
        digitalWrite(EN_Y, LOW);
    }
    f_ready = 0;
}

void setup(void) {
    ESP32PWM::allocateTimer(0);
    ESP32PWM::allocateTimer(1);
    ESP32PWM::allocateTimer(2);
    ESP32PWM::allocateTimer(3);
    Serial.begin(115200);
    WiFi.mode(WIFI_AP);
    WiFi.softAP(ssid, password); // configure ssid and password for softAP
    IPAddress IP = WiFi.softAPIP();
    //WiFi.begin(ssid, password); //begin WiFi connection
    Serial.println("trying to connect");
    // Wait for connection
    /*while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }*/
    Serial.println("");
    Serial.print("Connected to ");
    Serial.println(ssid);
    Serial.print("IP address: ");
    Serial.println(WiFi.localIP());
    stepper1.setPinsInverted (
        true, false, false
    );
    pinMode(EN_X, OUTPUT);
    digitalWrite(EN_X, HIGH);
    pinMode(EN_Y, OUTPUT);
    digitalWrite(EN_Y, HIGH);
    pinMode(STOP_X, INPUT);
    pinMode(STOP_Y, INPUT);
    trigger.setAcceleration(300000000000); //maximun is 3.4*10^38
    trigger.setPinsInverted(false, true);
    // trigger.setMinPulseWidth(100);
    stepper1.setMaxSpeed(MAXSPEED);
    stepper1.setAcceleration(MAXACCEL);
    stepper2.setMaxSpeed(MAXSPEED);
    stepper2.setAcceleration(MAXACCEL);
    myservo.setPeriodHertz(50); // standard 50 hz servo
    myservo.attach(servoPin, 1000, 2000); // attaches the servo on pin 18 to the servo object
    server.on("/", []) {
        server.send(200, "text/html", index_html_str);
    };
    server.on("/move", []) {
        if (f_ready) {

```

```

        Serial.println("move_positions:");
        int positions[5] = {0, 0};
        for (int i = 0; i < server.args(); i++) {
            positions[i] = server.arg(i).toInt();
            Serial.println(positions[i]);
        }
        prepareRelativeMove(positions);
        server.send(200, "text/html", "ok");
    }
    else {
        Serial.println("attempted move");
        server.send(200, "text/html", "wait");
    }
    delay(400);
    server.send(200, "text/html", "end");
});
server.on("/moveTo", []() {
    if (f_ready) {
        Serial.println("moveTo_positions:");
        int positions[5] = {0, 0};
        for (int i = 0; i < server.args(); i++) {
            positions[i] = server.arg(i).toInt();
            Serial.println(positions[i]);
        }
        prepareAbsoluteMove(positions);
        server.send(200, "text/html", "ok");
    }
    else {
        Serial.println("attempted move");
        server.send(200, "text/html", "wait");
    }
});
server.on("/home", []() {
    if (f_ready) {
        int positions[5] = {100000, 0};
        prepareRelativeMove(positions);
        server.send(200, "text/html", "ok");
        homing = true;
    }
    else {
        Serial.println("attempted move");
        server.send(200, "text/html", "wait");
    }
});
server.on("/trig", []() {
    if (trigger.isRunning()) {
        server.send(200, "text/html", "wait");
    }
    else {
        if (server.arg(1).toInt()) {
            trigger.setMaxSpeed(server.arg(1).toInt());
        }
        else {
            trigger.setMaxSpeed(Default_trig_speed); //default = 100 hz
        }
        if (!server.arg(0).toInt()) {
            trigger.move(1);
        }
        else {
            trigger.move(server.arg(0).toInt());
        }
        server.send(200, "text/html", "ok");
    }
}

```

```

});
server.on("/trig_begin", []() {
  if (server.arg(0).toInt()) {
    trigger.setMaxSpeed(server.arg(0).toInt());
  }
  else {
    trigger.setMaxSpeed(Default_trig_speed); //default = 100 hz
  }
  //delay(2);
  // trigger.stop();
  delay(2);
  trigger.move(biggestlong);
  server.send(200, "text/html", "ok");
});

server.on("/trig_stop", []() {
  //trig_on = 0;
  trigger.stop();
  server.send(200, "text/html", "ok");
});

server.on("/trig_left", []() {
  server.send(200, "text/html", String(trigger.distanceToGo()));
});

server.on("/set_MaxSpeed", []() {
  MAXSPEED = server.arg(0).toInt();
  server.send(200, "text/html", "ok");
});

server.on("/set_MaxAccel", []() {
  MAXACCEL = server.arg(0).toInt();
  server.send(200, "text/html", "ok");
});

server.on("/set_currentPos", []() {
  int positions[5] = {0, 0, 0, 0, 0};
  for (int i = 0; i < server.args(); i++) {
    positions[i] = server.arg(i).toInt();
  }
  stepper1.setCurrentPosition(positions[0]);
  stepper2.setCurrentPosition(positions[1]);

  server.send(200, "text/html", "ok");
});

server.on("/get_ready", []() {
  if (f_ready) {
    server.send(200, "text/html", "ready");
  }
  else {
    server.send(200, "text/html", "wait");
  }
});

server.on("/get_pos", []() {
  server.send(200, "text/html", String(String(stepper1.currentPosition()) + "," +
String(stepper2.currentPosition())));
});

server.on("/stop", []() {
  stepper1.stop();

```



```

    stepper2.stop();
    server.send(200, "text/html", "ok");

});

server.on("/always_on", []() {
    alwaysOn = server.arg(0).toInt();
    server.send(200, "text/html", "ok");
    if (alwaysOn) {
        digitalWrite(EN_X, LOW);
        digitalWrite(EN_Y, LOW);
    }
    else {
        digitalWrite(EN_X, HIGH);
        digitalWrite(EN_Y, HIGH);
    }
});

server.on("/servo_on", []() {
    servoOn = server.arg(0).toInt();
    server.send(200, "text/html", "ok");

});
server.begin();
Serial.println("Web server started!");

xTaskCreatePinnedToCore(
    Task1code, /* Task function. */
    "Task1", /* name of task. */
    10000, /* Stack size of task */
    NULL, /* parameter of the task */
    1, /* priority of the task */
    &Task1, /* Task handle to keep track of created task */
    0); /* pin task to core 0 */
delay(100);

// xTaskCreatePinnedToCore(
//     Task2code, /* Task function. */
//     "Task2", /* name of task. */
//     10000, /* Stack size of task */
//     NULL, /* parameter of the task */
//     1, /* priority of the task */
//     &Task2, /* Task handle to keep track of created task */
//     1); /* pin task to core 1 */
// delay(100);

}

void Task1code( void * pvParameters ) {
    for (;;) {
        server.handleClient();
        if ( !stepper1.isRunning() && !stepper2.isRunning() ) {
            servoPos = servoDOWN;
            if (servoOn) {
                delay(200);
            }
            f_ready = 1;
            //server.send(200, "text/html", "move finished");
            if (!alwaysOn) {
                digitalWrite(EN_X, HIGH);
                digitalWrite(EN_Y, HIGH);
            }
        }
    }
}

```

```

    vTaskDelay(10);
    // if(trig_on){
    //   trigger.move(1);

    // }
  }
}
/*
void Task2code( void * pvParameters ){
  for(;;){

  }
}
*/
void loop(void) {
  stepper1.run();
  stepper2.run();
  trigger.run();
  myservo.write(servoPos);
  if (digitalRead(STOP_X) == LOW) {
    //Serial.println("bateu");
    if (homing) {
      stepper1.stop();
      stepper1.setCurrentPosition(0);
      homing = false;
    }
  }
  if (digitalRead(STOP_Y)) {
    stepper2.stop();
    //f_ready = 0;
  }
}

```

contents.h:

```

const char *index_html_str="<p>5 axis mechanical arm - version 1.0</p>\n\
<p>IP:&nbsp;http://192.168.4.1/</p>\n\
<p>Steps/degree:</p>\n\
<p>X: 192 &nbsp;&nbsp;Y
<p>Command list:</p>\n\
<p>.../move?=valueX&amp;=valueY&amp;=valueZ&amp;=valueA&amp;=valueC &nbsp;&nbsp;- relative
move, will return \"ok\" or \"wait\"</p>\n\
<p>.../trig</p>\n\
<p>.../set_MaxSpeed?=value - in steps/sec</p>\n\
<p>.../set_MaxAccel?=value - in steps/sec<sup>2</sup></p>\n\
<p>.../get_ready - will return \"ready\" or \"wait\"</p>\n\
<p>.../get_pos - will return \"x=value1 y=value2 z=.....\" current position in absolute steps (sinse
reboot)</p>\n\
<p>.../stop - stops as soon as possible (acording to deceleration) will always return \"ok\"</p>\n\
<p>...target_pos - will return\"x=value1 y=value2 z=.....\" the target position in absolute steps (sinse
reboot)</p>\n\
";

```

Apêndice B

Matlab® Script

Nessa subseção são apresentados dois scripts de Matlab® utilizados durante as varreduras. O primeiro, 'CNC_moveTo.m' é utilizado para requisitar um movimento para o manipulador. O segundo, 'CNC_zeroing.m' é utilizado somente quando é necessário corrigir a posição do manipulador, caso ele não seja inicializado em 0°.

CNC_moveTo:

```
function [o] = CNC_moveTo(ip_CNC,posXmm,posYmm,posZmm,posAmm,posCmm)

% Function will input all positions and only return when the CNC finished the movement[
    Stpmm = 120*1.6;
    %Stpmm = 400;
    StpDegree = 125;
    posX = -floor(Stpmm*posXmm); %ALTERADO DEVIDO AO LAYOUT DO TANQUE, DEVERÁ SER
    ALTERADO NO FIRMWARE POSTERIORMENTE
    posY = floor(Stpmm*posYmm);
    posZ = floor(Stpmm*posZmm);
    posA = floor(Stpmm*posAmm);
    posC = floor(Stpmm*posCmm);

    posXstr = num2str(posX);
    posYstr = num2str(posY);
    posZstr = num2str(posZ);
    posAstr = num2str(posA);
    posCstr = num2str(posC);

    srto = ['http://' ip_CNC '/moveTo?=' posXstr '&=' posYstr '&=' posZstr '&=' posAstr '&=' posCstr];
    srto2 = ['http://' ip_CNC '/get_ready'];
    s = 'wait';
    while(strcmp(s,'wait')) %loop para pedir o movimento
        pause(0.05);
        s = urlread(srto);
    end

    pause(0.1);
    %loop para conferir se o movimento foi terminado
    s = 'wait';
    while(strcmp(s,'wait'))
        pause(0.5);
        s = urlread(srto2);
    end
```

CNC_zeroing.m :

```
ip_CNC = '192.168.4.1';
CNC_moveTo(ip_CNC,4,0,0,0,0)%change the first number to the opposite current position.
s = 'wait';

while(strcmp(s,'wait'))
    pause(0.5);
    s = urlread(['http://' ip_CNC '/set_currentPos']);
end
```



CNC_zeroing.m:

```
ip_CNC = '192.168.4.1';  
CNC_moveTo(ip_CNC,4,0,0,0,0)%change the fist number to the oposite current position.  
s = 'wait';  
while(strcmp(s,'wait'))  
    pause(0.5);  
    s = urlread(['http://' ip_CNC '/set_currentPos']);  
end
```

Apêndice C

Arquivos SolidWorks®

Nessa subseção são apresentados os arquivos com as montagens (assembly) feitas utilizando SolidWorks® e as peças modeladas e impressas.

Montagem Mecânica:

Contém todas as peças utilizadas na montagem principal.

https://drive.google.com/file/d/1hP5U_QkMybW-nvh43VJ5-2Wgbupy1Mbw/view?usp=sharing

Peças contidas nessa montagem:

Peça superior da estrutura central.

Forma o a parte inferior do rolamento central. Prende os perfis de alumínio verticais da estrutura central e a caixa de redução do motor de passo.

<https://drive.google.com/file/d/1dw5TYDAjvypxydsuKvQpfE2pUJt33WpN/view?usp=sharing>

Bearing Holder

Peça responsável por manter as esferas de aço do rolamento espaçadas corretamente.

https://drive.google.com/file/d/1pIV_StqxapECyM-iiq_90p1503-ncbuW/view?usp=sharing

Peça central do braço

Peça que forma a parte superior do rolamento central e prende ao perfil de alumínio do braço.

<https://drive.google.com/file/d/1iopW3yJGTx8z2Z2sGYmaWpecSSKypVXp/view?usp=sharing>

Peça superior da estrutura periférica

Prende aos dois perfis verticais da estrutura periférica e desliza radialmente sobre o braço.

<https://drive.google.com/file/d/1ymaponbAwi5xbr3MWZ0vA0U9Hc6YNr7N/view?usp=sharing>

Pês da estrutura periférica

Aparafusam embaixo dos perfis verticais.

<https://drive.google.com/file/d/1jcgEhdnL5vpxAAqEk9ejmrtPdMLXrX6F/view?usp=sharing>

Peça de montagem do EMAT receptor.

Encaixa nos perfis da estrutura periférica, deslizando verticalmente. Possui furos para aparafusar o EMAT receptor e passar o barbante que prende ao servomotor.

https://drive.google.com/file/d/1bV35cqMi221Ciu_KcLiXrLWWQLnhvr3Z/view?usp=sharing

Mola (impressa) do EMAT receptor.

Pressiona o EMAT contra o corpo de prova.

<https://drive.google.com/file/d/1AFCJ75IJAup8i8ZXZqcT16GNaovqMGRP/view?usp=sharing>

Peça de montagem do servomotor.

É presa nos perfis verticais da estrutura periférica e ao servomotor.

<https://drive.google.com/file/d/1ZFZU6SIprqVjVAYA5bM7fHvEieI3prOz/view?usp=sharing>

Montagem Eletrônica

Contém a caixa e tampa da eletrônica, dispostos em relação aos componentes eletrônicos.

<https://drive.google.com/file/d/130dsFYbgpQ5Eow-BI5Oyi8KIhonmehpU/view?usp=sharing>

Caixa da eletrônica

Prende e protege todos os componentes eletrônicos

<https://drive.google.com/file/d/1qs38wL5vpkIDWTLBCwLYmNVIFHfSk3q1/view?usp=sharing>

Tampa da caixa da eletrônica

<https://drive.google.com/file/d/1ROhqNepeEFW0E3G1L-YWmWKLmQ5qCHm/view?usp=sharing>

Caixa de EMAT SHD1031S

Protege o EMAT e possui os pontos necessários para fixação.

<https://drive.google.com/file/d/1bzqPDzTwvdQPDlkr6kcnxzLAqCeDbV3Q/view?usp=sharing>

Caixa de EMAT SHD2531S

Protege o EMAT e possui os pontos necessários para fixação.

https://drive.google.com/file/d/1VOcID6WSKWmWbRi9Phv2r-Z_U9WH8F9g/view?usp=sharing